

AD-A058 676

NAVAL ACADEMY ANNAPOLIS MD
MICROPROCESSOR-BASED DIGITAL CONTROL OF ENVIRONMENTAL SYSTEMS.(U)

F/G 9/2

JUN 78 E D KEE

UNCLASSIFIED

USNA-TSPR-92

NL

1 of 2

AD
A058 676



AD A058676

DDC FILE COPY

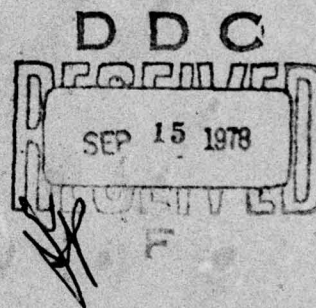
LEVEL II

(22)

**A TRIDENT SCHOLAR
PROJECT REPORT**

NO. 92

**"MICROPROCESSOR-BASED DIGITAL CONTROL
OF ENVIRONMENTAL SYSTEMS"**



**UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND
1978**

This document has been approved for public
release and sale; its distribution is unlimited.

78 09 14 039

14

USNA-TSPR-

22

U.S.N.A. - TSPR; no. 92 (1978)

9 Final Rept. 1977-1978

6

MICROPROCESSOR-BASED DIGITAL CONTROL
OF ENVIRONMENTAL SYSTEMS

A TRIDENT SCHOLAR PROJECT REPORT

by

10

Midshipman First Class Eddy D. Kee USN

Class of 1978

U. S. Naval Academy

Annapolis, Maryland

DDC
RECEIVED
SEP 15 1978
F

11 JUN 78

12 1480

E. E. Mitchell
Associate Professor E. E. Mitchell, Co-
Advisor, Weapons and Systems Engineering
Department.

Richard W. Kopka
Major R. Kopka, USAF, Co-Advisor, Weapons
and Systems Engineering Department.

Accepted for Trident Scholar Committee

Chairman

Chairman

1 JUN 78

Date

245600

LB

1

[illegible]

PREFACE

The United States and the entire world are approaching a point where energy consumption levels must be drastically reduced or there will be social, economic, and political changes the magnitude of which we have not seen since the great depression of the 1930's. Economists and energy analysts have placed the date for this impending crisis, by careful and conservative extrapolation, at about the year 2000. This is less than twenty-two years away from today. The massive shortages encountered during the oil embargo of 1973-1974, the rapidly rising cost of electrical power, gasoline, oil and coal, the recent coal strikes and their tremendous effects on the electrical power industry, and the ever-diminishing supplies of domestic oil and coal all point toward this crisis. I feel that it is the duty and obligation of every person in this country to make every effort to conserve and use properly the energy sources we now have available. The President of the United States expressed this need by declaring "the moral equivalent of war" on the impending crisis.

To fulfill my part in this campaign I decided to use my time and effort to develop a system which may save a significant percentage of the energy used to heat homes, buildings, and other inhabited structures. The Trident Scholar program gave me the opportunity to accomplish this task while completing my undergraduate education. Upon assuring myself that I could complete the courses in my major, Systems Engineering, as well as take the Humanities electives I consider essential to any undergraduate curriculum and take the

Trident Scholar program, I applied for it. My proposal was to design a microprocessor-based digital controller which could be adapted and programmed to control the heating of a house and to develop an energy saving strategy of programming to implement on the device.

The great difficulties which would have been encountered in controlling an actual house were overcome by developing a mathematical model of a house. The development of this model involved techniques which are used in devising the simulation of any physical system. The relationships between the variables of interest, the room temperatures, and their relationship to the input, the outside temperature, were determined. The methods used to find these relationships are general and may be applied to any mathematical model involving heat transfer and temperature levels. Some simplifying assumptions which did not greatly affect the accuracy of the model were made. These assumptions simplified the equations of the mathematical model considerably.

The finished model consisted of ten simultaneous, first order, differential equations. This model was programmed on the Dartmouth Time Sharing System using a general purpose, FORTRAN digital simulation program, DIGISIM, developed by E. E. Mitchell. This was done to get a quick and easy check on the model's validity. The initial runs revealed some errors in the coefficients causing the model to "blow up." These errors were corrected and the

78 09 14 039

equations were programmed on the EAI-681 Analog computer which, in conjunction with a PDP-15 minicomputer and associated digital-to-analog converters, analog-to-digital converters, and software, constituted the house simulation. The general nature of the simulation cannot be stressed enough. These simulation techniques can be applied to any building to determine such factors as the amount of energy saved by changing insulation, adding storm windows, and other modifications. These building modifications would require a large amount of money and time to be implemented in a real house. In this simulation they can be realized by simply re-computing the coefficients and changing potentiometer settings. The simulation has the disadvantage of being only a simulation and therefore having only a limited relation to reality. With care and good engineering approximations however, the results will closely simulate the response of the real structure. I feel that the development of this computer simulation was a very valuable side product of my Trident Scholar project and its use in developing more energy efficient structures has great potential. I welcome the use of this program or the theory used to develop it by anyone who might be pursuing the research of energy efficient house construction. I feel that a good deal of time and money could be saved through the use of this program.

In the development of the digital controller, a multitude of problems were encountered. I had almost no background in micro-electronics or microprocessors and their construction and programming.

After consulting many manuals and specification sheets and asking many questions I was able to accomplish the necessary tasks. A good deal of the answers to my questions and much help came from my advisors, Professor E. E. Mitchell and Major Richard Kopka. Bill Lowe and the hard working Technical Support Division gave me thousands of answers, hints, and parts as well as keeping the Hybrid Computer facilities ready and usable.

During the course of the year I have learned a tremendous amount about microprocessors, digital control, simulation, electrical engineering, thermodynamics, and other fields I would not have been exposed to otherwise. The formal courses I previously pursued in Sampled Data and Digital Control, Electrical Engineering, Thermodynamics, and Computer Simulation were all of great use and the knowledge I had retained from these courses was reinforced and supplemented by its application in the project. The Trident Scholar program was the most rewarding means of completing my four years at the Naval Academy, both in terms of doing my part in the war on excessive energy consumption and in preparing myself for service as an Officer in the Navy.

TABLE OF CONTENTS

Abstract	1
Preface	2
Table of Contents	6
Chapter 1 INTRODUCTION	7
Chapter 2 A MATHEMATICAL MODEL	10
Chapter 3 SIMULATION.	19
Chapter 4 THE INTERFACE.	21
Chapter 5 CONTROL.	26
Chapter 6 RESULTS.	28
Chapter 7 CONCLUSION.	35
Bibliography	40
Appendix A	41
Appendix B	55
Appendix C	75
Appendix D	97
Appendix E	108
Appendix F	123
Appendix G	144

CHAPTER 1

INTRODUCTION

The basic purpose of this project was the digital control of an environmental system such as a house or building where a specific environment must be maintained by the addition or subtraction of heat to individual rooms, groups of rooms, or the whole structure. Due to the possibly superfluous nature of air conditioning (cooling) and the possibility that its use will be curtailed in any reasonable emergency energy program, the controller has been designed to control only the heating of a structure. The use of this controller to control both the heating and cooling of a structure is entirely possible, with some additional programming it could be used satisfactorily to control the cooling of a structure. The control concepts to be considered are those involving a microprocessor controlled sampler and decision making device. The use of an actual structure to test and evaluate the controller was not feasible due to the typical failure of outside temperatures to follow test parameters and the large cost of rigging a building for alternate control by both digital controller and conventional thermostats. Therefore an analog/hybrid computer simulation of a building was used to test and develop the digital controller. The ease of varying input parameters, access to measurable outputs, and speed of test runs are some of the advantages of using an analog/hybrid computer simulation as opposed to the use of actual physical structures. Other benefits include the ability to actually model the entire control system, both the digital controller and the

conventional thermostat control on the digital portion of the hybrid computer. Thus the ideal performance of the digital controller can be measured before it is built, giving some indication of its applicability. The ability to develop a suboptimum programming routine for the microprocessor using the hybrid computer made the implementation of the digital controller much easier than if all programming had been developed on the microprocessor using the appropriate machine language.

The project was carried out in a series of stages, each building on the last and concluding with tests of the amount of energy that could be actually saved by using the controller. The first stage was the development of a hybrid computer simulation of a house and surrounding environment with respect to heat flow and temperature levels. Certain assumptions were made to simplify this simulation and these will be discussed in detail later. The second stage was to develop both conventional and digital control algorithms on the digital portion of the hybrid computer. The digital control algorithm was simplified for ease of implementation of the programs on the microprocessor, due to the difficult nature of machine language programming. The third stage of the project was to build the microprocessor-based computer and an interface device to connect the computer to the simulation. The microprocessor used was the Intel 8080, as configured in the Heath H-8 digital computer. The Heath H-8 was chosen because of its easily expandable memory, the interface capability, and the fact that this unit was easily obtained

and assembled by the user. The implementation of the digital controller would be within the grasp of the common user rather than limited to the skilled technician. The microcomputer was then linked up with the simulation and the control algorithms were programmed on the machine. The final stage of the project was to run many tests to evaluate the performance of the digital controller with respect to the amount of energy saved when compared to the energy used by conventional thermostat control of the heating of the structure. The performance of the device will give some indication of the feasibility of implementing it on existing houses and of building the control device into a new home.

CHAPTER 2

A MATHEMATICAL MODEL

The simplification of the control problem brought about by controlling a simulation rather than an actual structure led to the development of a mathematical model of a typical house. A five room, one story house with closets, two bedrooms, kitchen and living room was used as the modeled structure. A sloping roof and attic were also included in the house model (see Appendix A)

The materials used in the construction of walls, doors, windows, ceilings, and floors were all chosen as typical of many homes. In addition, the fact that this is a mathematical model made the rapid substitution of different materials possible, by substituting the new heat transfer coefficient into the governing equation. This enabled the effects of changing insulation to be quickly determined, and indications of the feasibility of adding or changing insulation in a house to be quickly obtained. The simulation was also easily changed to represent different types of houses, the only limiting factor being the number of non-contiguous rooms in the house. The analog computer used had a limited number of integrators and therefore could only be programmed to solve an equally limited number of differential equations. The possibility of using larger computers or slaving several of the computers together to achieve simulation of larger houses or buildings is obvious and bears further investigation.

At this point the basic governing equations were used to define

the heat flow and temperature conditions of each room. When this procedure is followed in great detail, taking in all things that might possibly affect the temperature in a room, the equations describing the temperature of a room become very large and increasingly difficult to program. To avoid this needless complication, and to get a clearer picture of the effects of the control on the variables of interest, certain simplifying assumptions were made. The first assumption made is that of constant air density. The value of the air density inside the house was assumed to be a constant value of .0764 pounds per cubic foot. The second assumption is that all doors and windows remain closed. This would be the case in very cold weather when doors would be opened for a minimum of time and windows would be kept closed. The third assumption is that of perfect ventilation and convection. The air in each room is assumed to be at a constant temperature throughout that room. The temperature of a room is the same regardless of the location within the room. The fourth assumption is that the infiltration effect is considered negligible, that no wind infiltrates through the walls or the windows.

All these assumptions are not really a factor in determining the results which are desired of the simulation. The goal is to control the heating of rooms in such a way that heat is conserved when compared with conventional methods of controlling the heating of houses. When assumptions are made which are typical of well built, heat conservative houses, the task of saving energy becomes a

more difficult procedure and the results will be more significant than the saving of energy in a low efficiency house would be. By modeling a house which is already energy efficient, the task of controlling the heating of this house in a manner which saves energy becomes a much more significant and important task.

The differential equations which relate the temperature of a room to the temperature of surrounding spaces is derived from two basic laws of heat flow. The first relates the heat flow across a heat conducting boundary, dQ/dt , to the coefficients of heat transfer in that boundary, k , the temperature differential across that boundary, dT , the thickness of the boundary, dX , and the area of the boundary, A (see Appendix A).

$$dQ/dt = dT/dX * k * A$$

The change of temperature of a thermal mass, in this case a room, is related to the thermal capacity of the material, $C_p * W$, and the heat flow into the material. W is the weight of the air in the room and C_p is the thermal constant of the air.

$$dT = 1/(C_p * W) * dQ$$

These equations relate the temperature inside a room to the temperature of the surrounding spaces. The method used to get the equations for each room in the house to be simulated is not trivial so the process will be considered in detail for one of the rooms. The heat transmission coefficients which combine the thermal conductivity of the materials, the thickness of the walls and the effects of such things as the film effect and other factors, are obtained from a table

of typical values for walls, doors, windows, floors, and ceilings.

See Appendix A for a table of the values used in this simulation.

These lumped coefficients allow the equation formulation process to proceed with much more ease than if the walls and other boundaries were taken material by material and the exact thermal transmission coefficient were obtained for each boundary. The living room (room A) is measured and areas of the interface with spaces adjacent are found. Heat flow is the product of the lumped transmission coefficient, u , the area of the wall, and the temperature differential across the wall. The interface with the outside consists of two walls, three windows, and a door. The respective areas are:

walls = 195 square feet

windows = 18 square feet

door = 19 square feet

When multiplied by the appropriate u factors the total coefficient of the outside temperature becomes:

$$\begin{aligned} \text{Heat flow} &= (T_{\text{out}} - T_a)(195 \text{ ft}^2 * .24 \text{ BTU/hr-ft}^2\text{-deg. F} + 19 \text{ ft}^2 \\ &* .48 \text{ BTU/hr-ft}^2\text{-deg. F} + 18 \text{ ft}^2 * 1.243 \text{ BTU/hr-ft}^2\text{-deg. F}) \end{aligned}$$

$$\text{Heat flow} = (T_{\text{out}} - T_a)(78.293 \text{ BTU/hr-deg. F})$$

The heat flow through the interface between room A and room E is computed.

$$\text{interior walls} = 10.33 \text{ ft}^2$$

$$\text{interior door} = 19 \text{ ft}^2$$

$$\begin{aligned} \text{Heat flow} &= (T_e - T_a)(10.33 \text{ ft}^2 * .33 \text{ BTU/hr-ft}^2\text{-deg. F} + \\ &19 \text{ ft}^2 * .64 \text{ BTU/hr-ft}^2\text{-deg. F}) \end{aligned}$$

$$\text{Heat flow} = (T_e - T_a)(15.57 \text{ BTU/hr-deg. F})$$

The heat flow through the interface between room A and room F is computed.

$$\text{interior walls} = 63.66 \text{ ft}^2$$

$$\text{interior door} = 19 \text{ ft}^2$$

$$\text{Heat flow} = (T_f - T_a)(63.66 \text{ ft}^2 * .33 \text{ BTU/hr-ft}^2\text{-deg. F} + 19 \text{ ft}^2 * .64 \text{ BTU/hr-ft}^2\text{-deg. F})$$

$$\text{Heat flow} = (T_f - T_a)(33.17 \text{ BTU/hr-deg. F})$$

The heat flow through the interface between room A and room H is computed.

$$\text{interior wall} = 29 \text{ ft}^2$$

$$\text{interior door} = 19 \text{ ft}^2$$

$$\text{Heat flow} = (T_h - T_a)(29 \text{ ft}^2 * .33 \text{ BTU/hr-ft}^2\text{-deg. F} + 19 \text{ ft}^2 * .64 \text{ BTU/hr-ft}^2\text{-deg. F})$$

$$\text{Heat flow} = (T_h - T_a)(21.73 \text{ BTU/hr-deg. F})$$

The heat flow through the interface between room A and room B is computed.

$$\text{interior wall} = 72 \text{ ft}^2$$

$$\text{Heat flow} = (T_b - T_a)(72 \text{ ft}^2 * .33 \text{ BTU/hr-ft}^2\text{-deg. F})$$

$$\text{Heat flow} = (T_b - T_a)(23.76 \text{ BTU/hr-deg. F})$$

The heat flow through the interface between the ceiling and room A is computed.

$$\text{area of ceiling} = 176 \text{ ft}^2$$

$$\text{Heat flow} = (T_{ci} - T_a)(176 \text{ ft}^2 * .12 \text{ BTU/hr-ft}^2\text{-deg. F})$$

$$\text{Heat flow} = (T_{ci} - T_a)(21.12 \text{ BTU/hr-deg. F})$$

The heat flow through the interface between the floor and room A

is computed.

$$\text{Area of floor} = 176 \text{ ft}^2$$

$$\text{Heat flow} = (T_{f1} - T_a)(176 \text{ ft}^2 * .58 \text{ BTU/hr-ft}^2\text{-deg. F})$$

$$\text{Heat flow} = (T_{f1} - T_a)(102.08 \text{ BTU/hr-deg. F})$$

The total heat flow into room A is the sum of all these individual heat flows, plus any heat source or sink(heater or cooler).

$$\begin{aligned} dQ/dt = & 78.294(T_{out} - T_a) + 15.57(T_e - T_a) + 33.17(T_f - T_a) + \\ & 21.73(T_h - T_a) + 23.76(T_b - T_a) + 102.08(T_{f1} - T_a) + 21.12(T_{ci} - T_a) \\ & + Q_{external}(\text{heater}) \end{aligned}$$

When all terms are multiplied out and the common terms are collected, the equation becomes:

$$\begin{aligned} dQ/dt = & -295.724 * T_a + 78.294 * T_{out} + 15.57 * T_e + 33.17 * T_f \\ & + 21.73 * T_h + 23.76 * T_b + 102.08 * T_{f1} + 21.12 * T_{ci} + Q_{ext}. \end{aligned}$$

This equation gives the heat flow into room A as a function of the temperatures of adjacent spaces and the applied heater. The units are BTU/hr. A quick check on the validity of the equation is to let all temperatures except that in room A be set equal to that in room A. When this is done, the heat flow equation becomes:

$$dQ/dt = Q_{ext}.$$

This equation is useful but what is required is a relation between the temperature in room A and the temperatures of adjacent spaces. To find this relation the thermal mass of room A must be computed. The thermal mass in the simulation is assumed to be only the air

within the room. This may not be an entirely accurate assumption, but the only thing affected by the changing of the thermal mass is the time constant of the room. The steady state response is not changed and the total amount of heat is not changed. We must now find the weight of the air in the room considered, room A. This is done by finding the volume of the room and multiplying it by the density of air.

$$\text{Volume of room A} = 176 \text{ ft}^2 * 8 \text{ ft} = 1408 \text{ ft}^3$$

$$W = \text{weight of the air} = .0746 \text{ lbs/ft}^3 * 1408 \text{ ft}^3 = 107.5712 \text{ lbs.}$$

$$1/CpW = (1/.24 \text{ BTU/lbs-deg F})(1/107.5712 \text{ lbs}) = .03873 \text{ deg F/BTU}$$

This factor relates the heat flow into room A to the temperature in room A. When the equation for the heat flow into room A is multiplied by this factor, the result is:

$$\begin{aligned} dT/dt = & -11.454 * T_a + 3.033 * T_{out} + .6031 * T_e + 1.285 T_f + \\ & .8417 * T_h + .9203 * T_b + 3.954 * T_{f1} + .8181 * T_{ci} + .03873 * Q_{ext}. \end{aligned}$$

(deg. F/hr)

This equation relates the temperature in room A to the temperature in adjacent spaces but the coefficients are not in the best form for programming on an analog computer so the change in temperature per minute is found by dividing the equation by 60.

$$\begin{aligned} dT/dt = & -.1909 * T_a + .0505 * T_{out} + .0101 * T_e + .0124 T_f + \\ & .014 * T_h + .0153 * T_b + .0659 * T_{f1} + .00646 * Q_{ext} + .0136 * T_{ci}. \end{aligned}$$

(deg F/min)

The same procedure is followed for all ten rooms to arrive at the complete model. There are ten compartment temperatures, five heaters,

and the outside temperature as variables in the equations. The outside temperature and the heaters are considered inputs and the temperatures are found by solving simultaneously the ten first order differential equations which comprise the model. See appendix A for a complete list of the equations and the procedure used for each room to arrive at these equations.

This model represents a frame house upon which certain restrictions have been placed. While this may not give a totally accurate picture of the characteristics of the house, the factors of interest are simulated with a high degree of correlation with the actual house, assuming the restrictions are true of the real house as well. The simulation which includes every variable and relation that could possibly affect the parameters of interest may very well be impossible to program. If it is possible, the program will be quite large and the resulting view of the parameters of interest will be fuzzy and probably inaccurate due to the accumulation of the small amounts of machine error, static charge, and other stochastic factors which are a part of every machine realized simulation. A simulation which neglects the minor relations and considers the major relations gives a clear, more easily realized view of the parameters in question. Even though the results may be slightly different from the real system, the use of computer simulation to accurately and quickly predict the performance of the actual system is a proven fact. This model of a house can be used to predict the behavior of an actual house which is exposed to varying outside temperatures. This prediction can be

used to design a control system for an actual house which will closely follow the behavior of the controller used to control the simulation of the house. The energy saved by using this controller in a real house should be roughly the same as that in the simulation, assuming the simulation is programmed with adequate engineering skill.

CHAPTER 3

SIMULATION

The simulation now consists of ten simultaneous differential equations with ten temperature variables, five heater inputs, an outside temperature input, and the floor temperature input. To solve these equations by hand would be very difficult unless the inputs were all assumed to be constants and the temperatures were also assumed to be constants. Then the derivatives would all be equal to zero and the solution becomes a matter of solving ten simultaneous algebraic equations. The answer to the problem of solving these equations in such a manner as to make the simulation useful is to mechanize the solution. That is, use the computer to solve the equations in a high speed mode, giving a nearly continuous solution. The entire model was programmed on the Dartmouth Time Sharing System using a general purpose digital simulation program, DIGISIM, developed by E. E. Mitchell. This solution gave some insight into the care originally exercised in computing the coefficients of the equations, as the simulation "blew up" when run. After recomputing the equations and correcting the errors, the house followed the outside temperature as it should, with a time lag dependent on the room. The controller was also included with this program and appeared to be functioning well. When the heat used over a period of time was measured with the rooms maintained at some temperature above the outside temperature, a change was noted when the delta time input to the program was changed. Attempts to make the delta time increment as small as was necessary to assure accurate solution by the Runge-Kutta method resulted in very

long run times for the simulation without assuring a good solution. The heat used over a period of time was seen to be a function of the time increment used to solve the Runge-Kutta integration algorithm. A listing of this program is included in Appendix B.

The simulation was moved to the analog computer where the dynamics were programmed on the EAI-681 analog computer and the controller and other programs were programmed on the PDP-15 digital computer. See Appendix C for the analog flow charts and listings of the digital programs used in the simulation.

CHAPTER 4

THE INTERFACE

To control the analog simulation of the house with the H-8 microcomputer, an interface between the H-8 and the EAI-681 was necessary. There were two primary reasons for this interface. The first was brought about by the fact that the H-8 performs operations on 8-bit binary numbers, while the EAI-681 uses analog voltage levels to represent variables in its solution of equations; they represent room temperatures in this case. To convert these various voltages into information which could be processed by the H-8 in order to control the simulation, an interface was needed. The second reason for an interface was the incompatibility of the H-8 control outputs and the voltages needed to control electronic switches on the EAI-681. Also the possibility of introducing a voltage level which is not of the type expected by the digital system, such as a negative voltage level existed. So a buffer system between the digital output of the H-8 and the digital portion of the EAI-681 was needed.

An 8-bit analog-to-digital converter was used to convert the analog voltage levels into binary, 8-bit numbers for the H-8 to process. There were several rooms to be monitored so there was a need for either several analog-to-digital converters or multiplexed input to a single analog-to-digital converter. Since the H-8 could only read one parallel I/O port at a time, receiving one 8-bit

number, then another, a multiplexer would have been necessary to select which analog-to-digital converter to read. To save space, time and materials, a multiplexer was used to select one signal to go into a single analog-to-digital converter which was read by a parallel I/O port on the H-8 when the "data valid" signal went low. The multiplexer was controlled by the output from one of the parallel I/O ports and the data from the analog-to-digital converter was read into the input side of the same port. To protect the analog-to-digital converter from the possible spurious voltage levels that might occur in the EAI-681, operational amplifier voltage followers were used as analog buffers on the inputs to the multiplexer and on the input to the analog-to-digital converter.

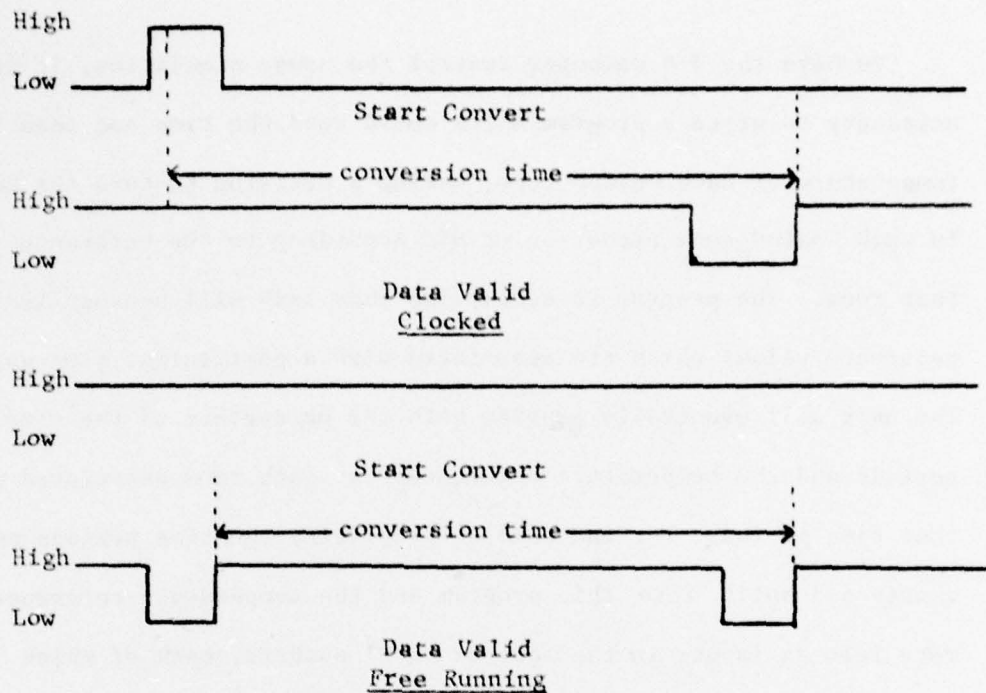
The basic component for the analog-to-digital interface was an ADC-EK8B monolithic analog-to-digital converter chip manufactured by Datel Systems Inc. (see specification sheet 1). The input for this chip came from a CD4051A analog multiplexer (see specification sheet 2) through a μ 741 operational amplifier voltage follower (see specification sheet 4). The multiplexer was controlled by TTL signals from the H-8 which were pulled up to the logic level needed to operate the CD4051A. These signals were pulled up to the necessary level by putting them through a N7407 non-inverting buffer/driver (see specification sheet 5). This buffer has open collector outputs, allowing the signals to be pulled up with a voltage source and a resistor for each signal. The output of the analog-to-digital converter chip was buffered with a CD4010AE

hex buffer/converter(see specification sheet 3). These signals then go into the parallel I/O port on the H-8 where they are read by the CPU. Control for the ADC-EK8B analog-to-digital converter was accomplished in an indirect manner. The signal which told the chip to "start convert" was connected to the "send data" output of a parallel I/O port on the H-8. This signal was high all the time except for a pulse of about five microseconds duration, occurring whenever the port is read by the CPU. This is because of the inverse logic used by Heath peripherals. Therefore, the analog-to-digital converter was essentially in a continuous mode(see signal relationship chart). The "data valid" signal from the analog-to-digital converter was sent via the CD4010AE buffer to the "data sent" input on the parallel I/O port. The hardware of the port insured that the port would not be read unless the signal present at the "data sent" input was making a transition from high to low. Since the chip was converting continuously, the changing voltage levels would be read at the value they were when the I/O port was read, not the value they were at when the "start convert" signal was sent to the converter chip. This helped eliminate some of the time lag involved in controlling the simulation.

The H-8 was programmed to use this interface in controlling the simulation. The control outputs were sent from the parallel I/O port where they were latched between output commands, into an N7407 buffer. On the output side of the buffer the signals were pulled up to the TTL logic level used in the EAI-681 analog computer,

positive five volts. The buffer also assured that the output signals would have the necessary fan-out strength. These signals were used to turn electronic switches on and off, thereby simulating the turning on and off of heaters.

The H-8 computer was able to control the simulation by taking a set of analog voltages and outputting a set of control signals, with the interface making the signals compatible both to the H-8 and to the EAI-681. Now the H-8 can be programmed to control the simulation without regard for the basic incompatibility of the signals. Ideally the interface should be transparent to the H-8 both in terms of input and output. This quality is achieved almost perfectly on the output of the H-8, but the inherent time lag involved with converting analog voltages into binary numbers make transparency of the input to the H-8 impossible. However with the ADC-EK8B converter in the continuous mode, the time lag is minimized.

ADC-EK8B Signal Relationship
Chart

The parallel port on the H-8 is set up to read the port when the "data sent" input makes a transition from high to low. Thus the ADC will be read when the data valid signal is making the transition from high to low. The start convert signal comes from the "send data" output on the H-8 parallel port. This signal is high except for a very short low pulse when the port is read due to the inverse logic of Heath peripherals. This signal is seen as an effectively continuous high signal by the ADC-EK8B, putting it in the free running mode.

CHAPTER 5

CONTROL

To have the H-8 computer control the house simulation, it was necessary to write a program which could read the time and then the temperature of each heated room, making a decision to turn the heater in each heated room either on or off according to the reference for that room. The program to accomplish this task will be supplied with reference values which are associated with a particular time period. The user will eventually program both the parameters of the time periods and the temperature references for each room associated with that time period. For the sake of simplicity the time periods were chosen and built into this program and the temperature references were left as inputs in the form of octal numbers, each of which corresponds to a particular voltage level or temperature level. It is necessary for the program to select the multiplexer command to put the time signal into the analog-to-digital converter of the interface. The value which results is used to select which time frame and therefore which set of temperature reference levels are used. The program then reads each room temperature in a similar manner, compares it to its reference level and adds the control word for that room to the command word or goes on to read the temperature of the next room. Since the control words are all in increasing powers of two, in binary numbers, they can simply be "ORed" together to give a total command word which will affect all heated rooms. This command word is then output to the control lines through a parallel input/output port, turning the heaters either on or off. The program

then cycles through again, starting with the read time operation. In real time the program would have a period of rest at this point, but the speed of the simulation made continuous operation necessary. The program flow chart, mnemonic code listing, octal operation code listing, mnemonic code definitions, and voltage level to octal conversion chart for the interface analog-to-digital converter are listed in Appendix E.

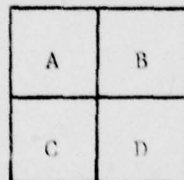
CHAPTER 6

RESULTS

The house simulation gave very satisfactory results. Its behavior compared favorably with the performance of an actual house. These conclusions about the simulation are subjective, as the house which was simulated does not exist and any comparison on a point by point basis was impossible. The simulation itself was used to develop an energy efficient programming strategy. In the project proposal it was assumed that a median temperature setting existed between the outside temperature and the comfortable inside temperature, which when used for the reference temperature for an unused room would yield a minimum heat use. This median temperature was sought by varying the reference temperature of one room while holding all others at an arbitrary temperature with heaters. The outside temperature was set at 20 degrees F and all inside rooms except the ones being tested, either one at a time or in combinations of two at a time, held at 75 degrees F. The rooms being tested were maintained at various temperatures ranging from the outside temperature, 20 degrees F, to the inside temperature, 75 degrees F. The simulation was then run for eight hours (equivalent real time) and the total heat used was measured. The results of these test runs are displayed graphically in Appendix F. These tests gave an indication of whether a temperature above the outside temperature would be a more heat efficient setting for the unused rooms. As the graphs indicate, the heat use is at a minimum when the temperature setting is below the steady state unheated temperature for the room. The steady state

unheated temperature for a room is that temperature which the room settles to when unheated. The difference between this temperature and the outside temperature is caused by the heat flow from adjacent rooms. In all cases the minimum heat use occurred when the unused room was unheated. The temperature that the room settled to when unheated is a function of the room, the temperature of the adjacent rooms, and the outside rooms. These results are in accordance with a theoretical analysis of the house.

A simplified model of a house with four symmetric, cubic rooms, no attic and all walls made of like materials is used for the theoretical analysis of the house. A simpler structure is used for this analysis because of the size and number of equations involved in the simulation used in this project.



$$dT(A)/dt = -K_A * T_a + K_2 * T_b + K_3 * T_c + K_5 * T_{out}$$

$$dT(B)/dt = -K_B * T_b + K_1 * T_a + K_4 * T_d + K_5 * T_{out}$$

$$dT(C)/dt = -K_C * T_c + K_1 * T_a + K_4 * T_d + K_5 * T_{out}$$

$$dT(D)/dt = -K_D * T_d + K_2 * T_b + K_3 * T_c + K_5 * T_{out}$$

Assume that rooms B, C, and D are heated, maintained at a temperature T_i , greater than T_{out} . Room A is unheated and the system is at steady state, so that all derivative terms are equal to zero.

$$T_a = (K_2 * T_1 + K_3 * T_1 + K_5 * T_{out})/K_A$$

The temperature in room A when unheated is shown to be a function of the temperatures in the adjacent rooms, the outside temperature, and the construction of the room. The heat use relations between the rooms can be determined in a similar manner. For the temperature change to be equal to zero, the net heat flow into the room has to be equal to zero. This means that the heater output is just equal to the heat loss of the room, either because of the size of the heater or more commonly the heater is turned off and on in such a manner that the total heat over a period of time is equal to the total heat loss of the room. To accomplish this a controller is used to maintain the temperature in the room at a constant level, thereby balancing the net heat flow at zero over some time period. To illustrate this the simplified model is again used. The heat flow equations for the rooms are:

$$dQ(A)/dt = -K_A' * T_a + K_2' * T_b + K_3' * T_c + K_5' * T_{out} + \text{Heater A} = 0$$

$$dQ(B)/dt = -K_B' * T_b + K_1' * T_a + K_4' * T_d + K_5' * T_{out} + \text{Heater B} = 0$$

$$dQ(C)/dt = -K_C' * T_c + K_1' * T_a + K_4' * T_d + K_5' * T_{out} + \text{Heater C} = 0$$

$$dQ(D)/dt = -K_D' * T_d + K_2' * T_b + K_3' * T_c + K_5' * T_{out} + \text{Heater D} = 0$$

When these equations are summed and the individual heaters are combined into a single heater the result is:

$$\begin{aligned} \text{Heater} = & (2 * K_1' - K_A') * T_a + (2 * K_2' - K_B') * T_b + (2 * K_3' - K_C') \\ & * T_c + (2 * K_4' - K_D') * T_d + 4 * K_5' * T_{out} \end{aligned}$$

The symmetry of the rooms make it possible to equate the coefficients of the room temperatures($(2 * K1' - KA') = (2 * K2' - KB') = Keq$)

$$\text{Heater} = Keq(Ta + Tb + Tc + Td) + 4 * K5' * Tout$$

This is the equation of a line which is a linear relation between the sums of the room temperatures and the heater value. Therefore any temperature setting which is greater than the outside temperature and also greater than the steady state temperature of the room being heated will increase the total amount of heat used. Since the simulation considered is only being heated and cooling is not a part of the energy consideration, the plot of this equation will be a straight line with slope of zero which turns upward with a positive slope dependent of the temperature of the unused room. The graphs in Appendix F illustrate this for the case when the room is maintained at the same temperature for an eight hour period. If cooling were considered in the model, the energy curve would be at a minimum when the temperature reference was equal to the steady state unheated temperature and would increase when the reference temperature was lowered because of the energy needed to cool the room to this lower temperature. It can also be seen that the constants Keq and $K5'$ are equal and opposite in sign, since the heater value will be equal to zero when all rooms are at the outside temperature.

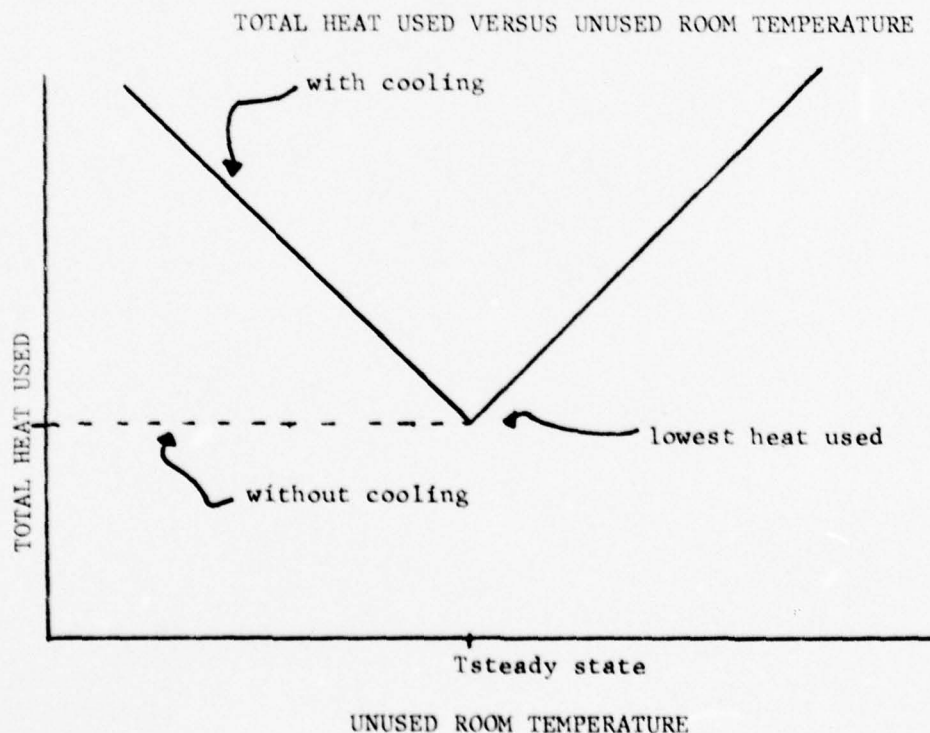
This analysis shows that for the steady state case the optimum temperature setting for an unused room is equal to or below the steady state unheated temperature of the room. Therefore a room which is to be unused for a period of time should be unheated for the greatest

saving of energy. The only consideration is the room itself, whether it can be allowed to go to the low temperatures that may result when the outside temperature is very low. If this is a factor the room may have to be programmed to stay above some low temperature, not the most efficient setting but necessary for the room. It has been stressed that this analysis is valid only for the steady state case. If the room temperatures are changing, the derivative terms do not go to zero and the solution becomes a great deal more complicated. The solution of this problem is one of the reasons that the simulation was implemented on the hybrid computer.

Tests were run for a variety of different conditions where a steady state solution would not be appropriate. The first series of tests involving the setting of the reference temperature to a constant level and maintaining it there for a period of eight hours were duplicated with some changes. The unused room or rooms were initially heated to the same level as the used rooms and then turned to a lower reference temperature for a period of time less than the total run time. The unused room was again turned back up to the reference temperature of the used rooms before the test run was ended. Two different time periods were used for the tests, one hour and about twenty minutes. The reference temperature of the unused room during these short time periods was again swept through values between the outside temperature and the temperature of the used rooms. The results of these runs are displayed graphically in Appendix F. As the graphs show, the results are much the same for the non-steady state

case as for the steady state case. The curves are less linear but the trend is upward with increasing reference temperature. The graph for the unused time of twenty three minutes shows that the rooms do not have time to cool off during this short time period, so the amount of heat saved is very little.

The proposal of finding a median temperature which would be used for the reference temperature for the unused rooms in order to effect a decreased heat usage when compared with other temperature settings for these rooms has been thoroughly investigated. A graph of the heat use equation developed for the simplified model will illustrate this temperature. Here we assume heating and cooling to show the optimum curves.



The most energy was saved by setting the unused room temperature to some value below the steady steady state temperature in that room. This chapter has given some analytical reasons why the controller was programmed to work the way it was. It has been shown that the best temperature setting for a room that is not to be used for a period of time greater than an hour is somewhere below the steady state temperature for that room. If the contents of the room make it necessary to maintain that room above the steady state temperature then the lowest setting possible is the most conservative of energy.

CHAPTER 7

CONCLUSION

In Chapter 6 certain conclusions were reached concerning the temperature setting of an unused room. This temperature setting was determined to be any temperature lower than the steady state temperature. If other constraints enter the problem, such as the need to keep the contents of the room above the freezing level or at some other temperature which the user desires, the lowest temperature consistent with these constraints is the most energy efficient for the unused room. Since these constraints as well as the decision about which rooms will be used and the time periods of this use must be determined by the person or persons who live in the house, a model cannot be made of the use of rooms. However, by taking what are typical work, sleep, and other activity patterns and making several use charts, some conclusions may be reached concerning the amount of energy saved by these use habits. Two typical use charts were made (see Appendix G) and used to write programs for the controller.

The first use chart is made to represent the use habits of a working couple whose children attend school all day. The house will be unused from the time the family all leave until they return in the afternoon. The bedrooms are not used until the evening but the other rooms are. After retiring, the living room and kitchen

are not used again until the early morning. This use chart can be programmed with four time intervals and the appropriate temperature settings for each room. It is assumed that the lowest temperature to which an unused room can cool is 35 degrees F. The rooms that are being used are set to 75 degrees F, and the outside temperature is at 20 degrees F. This set of parameters was programmed on the H-8 and used to control the house simulation for an equivalent real time of 24 hours. The total heat used is then compared with the heat used to heat the house for 24 hours with all rooms being heated to 75 degrees F for the entire period, also with an outside temperature of 20 degrees F. For this use chart a savings of 48.6 % was realized. This represents a substantial saving of both energy and money used to purchase that energy. The model's inherently incorrect results are effectively cancelled by examining the net percentage saving.

The second set of use habits are somewhat less saving of energy. The case of having someone who does not leave the house is considered in this use chart. At night the bedrooms and the bathroom are heated and the rest are unheated. During the day the opposite is true, all rooms except the bedrooms are heated. When programmed and compared in the same manner as the first use chart, a savings of 31.9 % was realized. These tests show that the controller does perform its task in a spectacular way. For a house such as the one modeled here the money saved by a 30 % reduction in heating costs spread over a winter such as that in 1976-1977 would be

quite a substantial sum. This alone would suggest that the digital controller be adopted by anyone whose house is configured in such a way (all electric, each room heated separately) that the installation costs are low. The builder of a new home would also be able to use the controller, the fixed costs of installation being cut down by the inclusion of the controller in the original house during construction. There are other ways that the controller and the simulation can be used.

The practice of lowering the setting of thermostats was one means of lowering the use of energy in homes. How efficient was this practice? To find out, the simulation was set up in a manner similar to the tests run for the use charts. In this case the rooms were all heated to the same temperature during the entire period. When the amount of heat used over a 24 hour period with the thermostats set at 65 degrees F is compared to the reference heat use, that of 75 degrees over a 24 hour period, a savings of 17.3 % is realized. The 10 degree difference is a little extreme but the test shows that a significant amount of energy is saved. Another application of the simulation is the measuring of the heat saving affected by adding insulation. To change the insulation in the simulation, the governing equations are changed to reflect the new thermal transmission coefficients in the walls. The changes are reflected in the analog simulation in the changing of potentiometer settings. If a large building were being simulated,

a computer could be programmed to change the potentiometer settings with an input of insulation values. A program which takes as an input the values of the insulation, computes new equation coefficients, computes new potentiometer settings and then sets the potentiometers could be easily written and implemented on the digital portion of a hybrid computer. A parameter sweep involving the input of a minimum and maximum value of insulation for walls, for ceilings, for roofs, and for windows could be written. This program would check the heat used by every combination of insulation values and reach what would be an optimum insulation configuration. Common sense tells us that this would be the thickest insulation everywhere, but perhaps this is not the case. The interior walls for example could be either totally insulated or uninsulated without changing the interface to the outside at all. The study of the most efficient way to heat our buildings is one of the most important ways to help stop the impending energy crisis. The heating of living spaces is the one single major use of heat that cannot be curtailed. People must have a livable environment or they will freeze to death. The use of digital controllers in homes will save some of the energy now used to heat homes and the use of the house simulation could be used to devise insulation and construction methods to save even more energy.

Yet another advantage to using the digital controller is the possibility of using the controller in a time-share mode to control

other things in the house. Since the rooms need to be sampled only about once every several minutes and the decision to turn the heater on or off takes approximately a few microseconds the microprocessor is idle for a large percentage of the time. This idle time could be used to control such things as the lights in the house, the burglar and fire alarms, the operation of various appliances, and many other specialized tasks. The use of the controller to control the peak power load of small industry is another possibility. A program which monitored the electrical load and made decisions about whether a machine could be turned on at that time or at a later time could keep the peak power level below the rate cut-off and save the company money. A central microcomputer for a house which controlled the temperature in individual rooms, the running of appliances, and many other tasks in the home could be the trend of the future. As the microprocessor industry becomes more well developed and the prices go down, this universal computer control concept may become a reality. The dreams of the early computer pioneers have been surpassed again and the machines which were once the province of the scientist are now the tools of the common man. Such uses as have been investigated here, the control of the temperature in a house for energy conservation, will become commonplace in years to come.

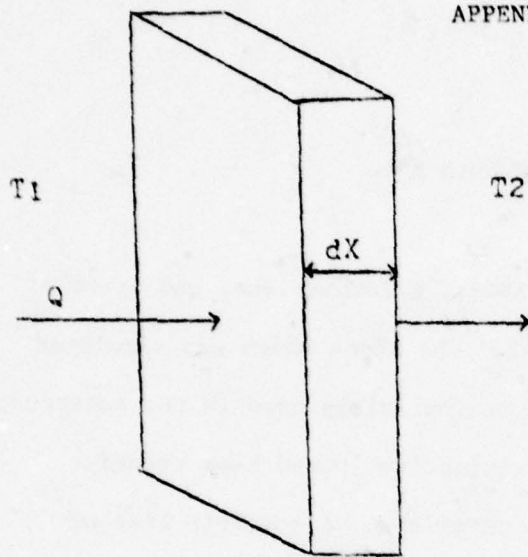
BIBLIOGRAPHY

- ADC-EK8B SERIES. Canton, Mass.: Datel Systems Inc. 1977.
- Basics of Parallel Hybrid Computers. Electronic Associates Inc., 1972. Publication number 00 800.3039-0-September 1972.
- Blum, Joseph J. Introduction to Analog Computation. New York: Harcourt, Brace and World, Inc., 1969.
- COS/MOS Digital Integrated Circuits. RCA Solid State '74 Databook Series, RCA Solid State. Somerville, N. J., 1974.
- EAI-681 Hybrid Laboratory Reference Handbook. United States Naval Academy: Department of Weapons and Systems Engineering.
- Jennings, Burgess H. Environmental Engineering, Analysis and Practice. New York: International Textbook Company, 1970.
- Jung, Walter G. IC OP-AMP Cookbook. Indianapolis: Howard W. Sams & Company, Inc., 1977.
- Kuo, Benjamin C. Discrete-Data Control Systems. Englewood Cliffs: Prentice-Hall, Inc., 1970.
- PDP-15 FORTRAN IV Language Manual. Maynard, Mass.: Digital Equipment Corporation, Number DEC-15-LFLMA-A-D, 1974.
- Perry, Oliver H., compiler. TTL Integrated Circuits Data Book. United States Naval Academy: Department of Weapons and Systems Engineering.

APPENDIX A

This Appendix contains the tables, illustrations, and lists of equations referred to in Chapter 2. The house which was simulated is drawn with side and end views, the materials used in the construction of this house are listed and the respective lumped heat transfer coefficients are assigned to the materials. A complete list of the simulation equations is also included, with each space being examined in detail with respect to the heat flow and temperature relationships to one another and to the outside temperature.

APPENDIX A



$$\frac{dQ}{dt} = kA \frac{dT}{dX}$$

Where

$\frac{dQ}{dt}$ = heat transfer per unit time (t);

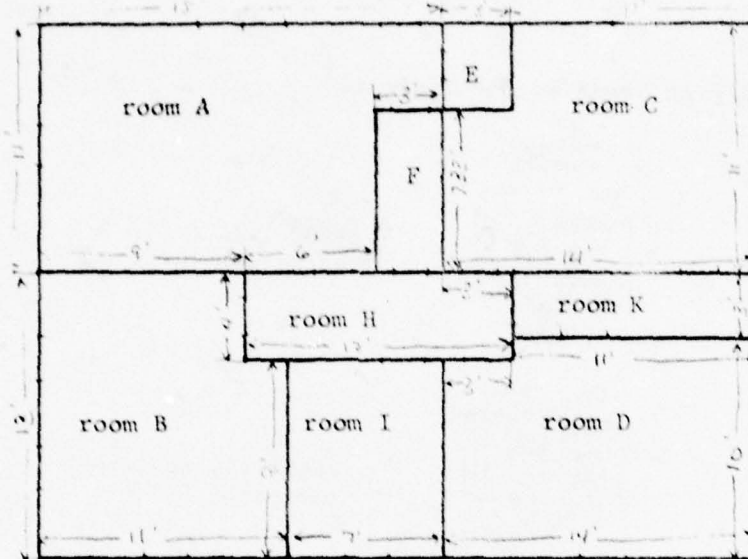
A = area of the section through which heat is flowing;

dT = temperature difference causing heat flow;

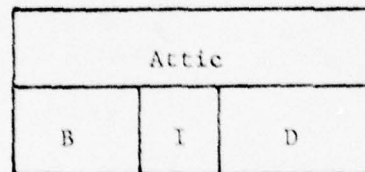
dX = length of path through which heat flows, measured in the direction of heat flow;

k = a proportionality factor called thermal conductivity.

APPENDIX A



Floor plan (from above)



side view



end view

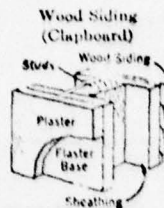
$$\text{Windows} = 6 \text{ ft}^2 \text{ (3ft x 2ft)}$$

$$\text{Doors} = 19 \text{ ft}^2$$

APPENDIX A

Lumped Heat Transfer Coefficients¹Exterior walls

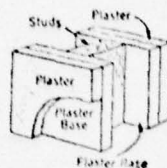
$$u = .24 \text{ BTU/hr-ft}^2\text{-deg F}$$



page 152, table 4-4
partition 5.C.

Interior walls

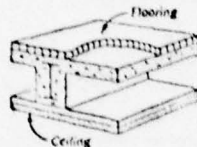
$$u = .33 \text{ BTU/hr-ft}^2\text{-deg F}$$



page 156, table 4-8
partition 5.B.

Floors

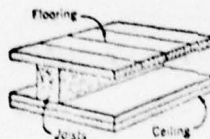
$$u = .58 \text{ BTU/hr-ft}^2\text{-deg F}$$



page 158, table 4-11
floor 2.C.

Ceiling

$$u = .12 \text{ BTU/hr-ft}^2\text{-deg F}$$



page 157, table 4-10
ceiling 5.E.

Roof

$$u = .55 \text{ BTU/hr-ft}^2\text{-deg F}$$



page 161, table 4-14
roof 1.I.

Inside Doors $u = .64 \text{ BTU/hr-ft}^2\text{-deg F}$

page 163

Outside Doors $u = .48 \text{ BTU/hr-ft}^2\text{-deg F}$

page 163

Windows $u = 1.243 \text{ BTU/hr-ft}^2\text{-deg F}$

page 164

¹B. H. Jennings, Environmental Engineering, Analysis and Practice (New York: International Textbook Company, 1970), pp. 152-164.

APPENDIX A

ROOM A

INTERFACE	AREA(ft ²)	COEFFICIENT
A to Outside	232 (1 door, 3 windows)	78.294
A to E	29.333 (1 door)	15.57
A to F	82.666 (1 door)	33.17
A to H	48. (1 door)	21.73
A to B	72	23.76
A to Floor	176	102.08
A to Ceiling	176	21.12

$$\begin{aligned}dT/dt \text{ (deg. F/min)} = & -.1909 * T_a + .0506 * T_{out} + .0101 * T_e + \\& .0124 * T_f + .014 * T_h + .0153 * T_b + .0659 * T_{f1} + .0136 * T_{ci} \\& + .0006455 * \text{Heater A}\end{aligned}$$

APPENDIX A

ROOM B

INTERFACE	AREA(ft ²)	COEFFICIENT
B to Outside	192 (1 door, 2 windows)	62.676
B to A	72	23.76
B to H	48 (1 door)	21.73
B to I	72	23.76
B to Floor	135	78.3
B to Ceiling	135	16.2

$$\begin{aligned} DT/dt \text{ (deg.F/min)} = & -.1905 * T_b + .0527 * T_{out} + .02 * (T_a + T_i) \\ & + .0183 * T_h + .0659 * T_{f1} + .0136 * T_{ci} + .0008416 * \text{Heater B} \end{aligned}$$

APPENDIX A

ROOM C

INTERFACE	AREA(ft ²)	COEFFICIENT
C to Outside	176 (2 windows)	54.276
C to E	53.33	17.6
C to F	58.66	19.36
C to H	24 (1 door)	13.81
C to K	88 (1 door)	34.93
C to Floor	143	82.94
C to Ceiling	143	17.16

$$\begin{aligned}dT/dt \text{ (deg. F/min)} = & -.1907 * T_c + .014 * T_e + .0153 * T_f + \\& .011 * T_h + .0278 * T_k + .0431 * T_{out} + .0659 * T_{f1} + .0136 * \\& T_{ci} + .0007945 * \text{Heater C}\end{aligned}$$

APPENDIX A

ROOM D

INTERFACE	AREA(ft ²)	COEFFICIENT
D to Outside	192 (2 windows)	58.116
D to K	88 (1 door)	34.93
D to H	72	23.76
D to Floor	137	79.46
D to Ceiling	137	16.44

$$\begin{aligned}dT/dt \text{ (deg. F/min)} = & -.1900 * T_d + .029 * T_k + .0197 * T_i \\& + .0136 * T_h + .0482 * T_{out} + .0659 * T_{f1} + .0136 * T_{ci} \\& + .0008293 * \text{Heater D}\end{aligned}$$

APPENDIX A

ROOM E

INTERFACE	AREA(ft ²)	COEFFICIENT
E to Outside	24	5.76
E to C	53.333	17.6
E to A	29.333 (1 door)	15.57
E to Floor	11	6.38
E to Ceiling	11	1.32

$$\begin{aligned}dT/dt \text{ (deg. F/min)} = & -.4816 * T_e + .1818 * T_c + .1608 * T_a \\& .0595 * T_{out} + .0659 * T_{fl} + .0136 * T_{ci}\end{aligned}$$

APPENDIX A

ROOM F

<u>INTERFACE</u>	<u>AREA(ft²)</u>	<u>COEFFICIENT</u>
F to A	82.666 (1 door)	33.17
F to C	58.666	19.36
F to H	24 (1 door)	13.81
F to Floor	22	12.76
F to Ceiling	22	2.64

$$\begin{aligned} dT/dt \text{ (deg. F/min)} = & -.4221 * T_f + .1713 * T_a + .10 * T_c + \\ & .0713 * T_h + .0659 * T_{f1} + .0136 * T_{ci} \end{aligned}$$

APPENDIX A

ROOM H

INTERFACE	AREA(ft ²)	COEFFICIENT
H to A	48 (1 door)	21.73
H to F	24 (1 door)	13.81
H to C	24 (1 door)	13.81
H to K	24 (1 door)	13.81
H to D	32 (1 door)	16.45
H to I	56 (1 door)	24.73
H to B	48 (1 door)	21.73
H to Floor	48	27.84
H to Ceiling	48	5.76

$$\begin{aligned}dT/dt \text{ (deg. F/min)} = & -.3771 * T_h + .05143 * (T_a + T_b) + \\& .0327 * (T_f + T_c + T_k) + .0389 * T_d + .0577 * T_i + .0659 * T_{f1} \\& + .0136 * T_{ci}\end{aligned}$$

APPENDIX A

ROOM I

<u>INTERFACE</u>	<u>AREA (ft²)</u>	<u>COEFFICIENT</u>
I to Outside	56 (1 window)	19.458
I to B	72	23.76
I to H	56 (1 door)	24.37
I to D	72	23.76
I to Floor	63	36.54
I to Ceiling	63	7.56

$$\begin{aligned} dT/dt \text{ (deg.F/min)} = & -2443 * T_i + .0429 * T_b + .044 * T_h + .0428 \\ & * T_d + .0351 * T_{out} + .0659 * T_{fl} + .0136 * T_{ci} + .001804 * \\ & \text{Heater I} \end{aligned}$$

APPENDIX A

ROOM K

<u>INTERFACE</u>	<u>AREA (ft²)</u>	<u>COEFFICIENT</u>
K to Outside	24	5.76
K to D	88 (1 door)	34.93
K to H	24 (1 door)	13.81
K to C	88 (1 door)	34.93
K to Floor	33	19.14
K to Ceiling	33	3.96

$$\begin{aligned} dT/dt \text{ (deg. F/min)} = & -.3874 * T_k + .0198 * T_{out} + .1203 * T_d \\ & + .0475 * T_h + .1203 * T_c + .0659 * T_{f1} + .0136 * T_{ci} \end{aligned}$$

APPENDIX A

ATTIC

INTERFACE	AREA(ft ²)	COEFFICIENT
Attic to Outside	886.81 (roof) + 166.28 (walls)	527.652
Attic to A	176	21.12
Attic to B	135	16.2
Attic to I	63	7.56
Attic to D	137	16.44
Attic to K	33	3.96
Attic to H	48	5.76
Attic to F	22	2.64
Attic to E	11	1.32
Attic to C	143	17.16

$$\begin{aligned}
 dT/dt \text{ (deg. F/min)} = & -.2118 * T_{ci} + .007215 * T_a + .00553 * T_b \\
 & + .002583 * T_i + .00562 * T_d + .001353 * T_k + .001968 * T_h \\
 & + .00902 * T_f + .000451 * T_e + .005863 * T_c + .1803 * T_{out}
 \end{aligned}$$

APPENDIX B

This Appendix contains a listing of the general purpose digital simulation program, DIGISIM, developed by E. E. Mitchell. This program is configured to simulate the model of a house used in this project and is set up to print out a list of numbers corresponding to the temperature in rooms A and C and the total heat used over the time period, not specified in the listing. This program was of great use in removing the errors in the model, since the digital simulation integration technique is sensitive to small errors in the model and these small errors caused the simulation to "blow up." By running the model again and again until the temperatures were maintained at a steady value with no heater inputs, the errors were removed.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDG

DIGISIM

```

***** DIGISIM --- DIGITAL SIMULATOR -- E.E. MITCHELL 2/77
*
*** REMEMBER -- RUN FORTRAN
*
* LINES 400-650:    INITIAL TIME DATA BLOCK
* LINES 750-1000:   SYSTEM DYNAMIC EQUATIONS
* LINES 1200-1500:  OUTPUT STATEMENTS
* LINES 1700-1850:  TERMINAL CALCULATION REGION
*
* BASIC DATA: TMAX, DELT, NSAMPL
* BASIC STATEMENTS: X=ENTGRL(XD,X(0)), CALL OUTPUT(X,"X")
* ADDITIONAL FUNCTIONS: STEP, FRSTOR, SCNDOR, PULSE, SQUARE, DELAY, DIGTAL, LIMIT
*                      PTRAIN
* FOR MORE DETAILS CONTINUE LISTING
*
**** INITIAL TIME REGION
*
* THIS PROGRAM HAS AN INITIAL TIME DATA REGION EXTENDING FROM LINES 400-750
* THIS SPACE IS USED TO DEFINE CONSTANTS, INITIAL VALUES AND OTHER NON
* CHANGING PARAMETERS REQUIRED BY YOUR PROGRAM. ALSO INITIAL CALCULATIONS
* MAY BE MADE HERE (IF YOU DEFINE THE REQUIRED PARAMETERS).
* IN ADDITION, SPECIAL CONSTANTS FOR BUILT IN DIGISIM FUNCTIONS ARE DEFINED
* HERE. FOR INSTANCE, COMMONLY DEFINED TERMS IN THIS AREA ARE:
* HEAD=" " ONE LINE OF 80 OR LESS CHARACTERS BETWEEN THE QUOTES.
*          THIS IS TYPED OUT AS A HEADING FOR THE RUN
* DELT     INTEGRATION STEP SIZE (DELT=0.05 DEFAULT)
* TMAX     TOTAL SIMULATED RUN TIME (TMAX=3 DEFAULT)
* NSAMPL   OUTPUT SPACING- ANSWERS PRINTED EVERY DELT*NSAMPL
*          TIME INTERVAL (NSAMPL=2 DEFAULT)
* ZK       DIGITAL CONTROLLER GAIN
* ZA(1-4), ZB(1-4) DIGITAL CONTROLLER COEFFICIENTS (ZA()=ZB()=0 DEFAULT)
* TSAMP    DIGITAL CONTROLLER SAMP TIME (TSAMP =0 DEFAULT)
* TLAG     DELAY PERIOD FOR TIME DELAY (TLAG=0 DEFAULT)
* TO       INITIAL TIME IF OTHER THAN 0 (TO=0 DEFAULT)
* TKPLOT   IF TKPLOT=1 THE OUTPUT RESPONSE IS WRITTEN INTO A FILE
*          SAVED BY YOU. IT IS IN THE PROPER FORM FOR PLOTTING
*          ON THE TEKTRONIC GRAPHIC TERMINALS. (TKPLOT=0 DEFAULT)
* PPLANE   IF PPLANE=1 THE TKPLOT FILE IS WRITTEN TO MAKE A PHASE
*          PLANE PLOT OF THE FIRST TWO CALL OUTPUT VARIABLES
*
**** DYNAMIC REGION
*
* THE PROGRAM HAS A DYNAMIC REGION FROM STATEMENT NUMBERS 750-1000. IN THIS
* AREA THE SYSTEM DYNAMICS, CONSISTING OF DIFFERENTIAL AND ALGEBRAIC
* EQUATIONS, ARE DEFINED.
* IN THE DYNAMIC REGION, THE FULL REGULAR FORTRAN LIBRARY IS AVAILIABLE.
* FOR THAT MATTER, IT IS AVAILIABLE ANYWHERE IN THE PROGRAM. IN ADDITION
* THE FOLLOWING FUNCTIONS ARE FURNISHED: (X=INPUT, Y=OUTPUT)
*
* T = TIME

```

DIGISIM (continued)

```
* Y = ENTGRL(X,XO) -- XO=INITIAL VALUE - DECIMAL POINT REQUIRED IF NUMBER
* Y = STEP(T1) -- Y=0 FOR T<T1, Y=1 FOR T>=T1
* Y = PULSE(T1,T2) -- Y=0 EXCEPT Y=1 WHEN T1<=T<=T2
* Y = SQUARE(T1,T2,T3,P) -- SQUARE WAVE WITH AMPLITUDE OF Y=1. LEADING EDGE
* OF FIRST PULSE AT T3, PULSE WIDTH IS T1 TO T2,
* REPEATS EVERY P SECONDS.
* Y = FRSTOR(X,A,B,C,D) -- FIRST ORDER TRANSFER FUNCTION DEFINED AS
* (A*S+B)/(C*S+D) C IS NOT ZERO
* Y = SCNDOR(X,A,B,C,D,E,F) -- SECOND ORDER TRANSFER FUNCTION DEFINED AS
* (A*S*S+B*S+C)/(D*S*S+E*S+F) D IS NOT ZERO
* Y = DELAY(X,TLAG) -- TIME DELAY, Y = X AFTER X IS DELAYED TLAG SECONDS
* Y = DIGTAL(X) -- DIGITAL FILTER OR CONTROLLER SIMULATOR. DELT SHOULD BE
* <=TSAM/10. TSAM=SAMPLE INPUT-OUTPUT TIMES OF THE DIGITAL
* CONTROLLER. TRANSFER FUNCTION IS:
* (ZK*(ZA(1)*X+ZA(2)*X-1+ZA(3)*X-3) - ZB(2)*Y-1
* -ZB(3)*Y-2 )/ZB(1)
* X=CURRENT INPUT,X-1 = LAST INPUT,ETC. DITTO FOR Y
* Y = LIMIT(U1,X,U2) -- Y=U1 IF X<U1, Y=U2 IF X>U2, Y=X OTHERWISE
* Y = PTRAIN(T1) -- Y=1 IF T = N*T1; THIS IS A PULSE TRAIN, UNIT PULSE
* EVERY N*T1 SECONDS
```

**** OUTPUT REGION

```
* THE PROGRAM HAS AN OUTPUT REGION FROM STATEMENT NUMBER 1200-1500. IN THIS
* REGION THE VARIABLES TO BE PRINTED AND PLOTTED ARE DEFINED. TO OUTPUT
* X, XDOT AND X2DOT, THE STATEMENTS ARE:
```

```
* CALL OUTPUT(X,"X")
* CALL OUTPUT(XDOT,"XDOT")
* CALL OUTPUT(X2DOT,"ACCEL")
```

```
* ANY 8 CHARACTERS MAY BE USED BETWEEN THE " ", THESE ARE THE COLUMN
* HEADINGS OVER THE PRINTED VALUES.
```

```
** NOTE ** TIME IS OUTPUT AUTOMATICALLY
```

```
** NOTE ** THE FIRST 6 CALL OUTPUT VARIABLES ARE PLOTTED
```

**** TERMINAL CALCULATION REGION

```
* THE PROGRAM HAS A TERMINAL REGION FROM STATEMENT NUMBERS 1700-1850. THIS
* AREA IS USED TO MODIFY PARAMETERS, ECT. AT THE END OF A RUN. ONE CAN
* THEN GO BACK AND INTEGRATE AGAIN AND AGAIN. A TYPICAL PROBLEM THAT
* WOULD USE THIS AREA IS A BOUNDARY VALUE PROBLEM.
```

```
* IF YOUR PROGRAM CONTAINS THE STATEMENT
* GO TO 9000
* CONTROL IS TRANSFERED BACK TO THE TOP OF THE INITIAL AREA
* IN A LIKE MANNER, THE STATEMENT
* GO TO 9100
```

DIGISIM (continued)

* TAKES CONTROL TO THE BOTTOM OF THE INITIAL AREA, (TOP OF DYNAMIC AREA)

*

**** END OF INSTRUCTIONS

*

IMPLICIT REAL(I-N)

INTEGER ICZ, NICZ, ITZ, NITZ, ITMAX, KERR, IAZ, NZZ, KPZ, NSAMPL, KTZ, IIZ, JJZ, IZ

INTEGER ICNT, NVAR, IDIGL, ICN, IADCNT, ISZZ

COMMON ZOUT, XVZ(6, 402), ICNT, NVAR, T, TO, TMAX, DELT, HDELT, ICZ, NICZ, ITZ, NITZ,

& ITMAX, KERR, IAZ(100), Y1PZ(100), Y4PZ(100), X1PZ(100), X4PZ(100), NZZ, KPZ,

& NSAMPL, KTZ

COMMON /HOLD/ IADCNT, ITQA, JCNT, IQCNT, HLD(500)

COMMON /OUTP2/ IIZ, JJZ, AZZ, ISZZ, AMXZ(25), AMNZ(25), HEAD, NPLOT, TKPLOT,
& PPLANE

COMMON/DIGITA/IDIGL, ZA(4), ZB(4), TSAM, ZK, NCNT

COMMON/TEMP1/SWA, SWB, SWC, SWD, SWH, SWI

COMMON/TEMP2/TA, TB, TC, TD, TH, TI, TOUT

CHARACTER PDQ, PDQ2, IPQ, AZZ(25), HEAD*80

REAL LIMIT

INTEGER TKPLOT

DATA ICN, IIZ, JJZ/0, 0, 0/

ZOUT=ICNT=IDIGL=0

TKPLOT=0

NPLOT=6

DO 109 IZ=1, 4

ZA(IZ)=0.

109 ZB(IZ)=0.

ZK=0.

IADCNT=0

DO 108 IZ=1, 500

108 HLD(IZ)=0

1 DO 2 IZ=1, 100

2 IAZ(IZ)=0

9000CONTINUE

TO=0.

DELT=.05

NICZ=10

NITZ=4

**** INITIAL REGION *** LINES 400-650

*

*

*

*

*

*

DIGISIM (continued)

*
*
*
*

IF(TKPLOT .EQ. 0)GO TO 9100

WRITE(ZOUT,87)

87 FORMAT(" WHAT IS YOUR SAVED FILE NAME")

READ(ZOUT,14)IPQ

OPENFILE 3,IPQ

REWIND 3

ENDFILE 3

9100KTZ=0

T=TO

HDELT=0.5*DELT

KERR=0

ICZ=1

ITMAX=NICZ

4 ITZ=1

5 NZZ=0

**** DYNAMIC REGION *** LINES 750-1000

TED=-.4816*TE+.1818*TC+.1608*TA+.0595*TOUT+.0659*TFL+.0136*TCI

TFD=-.4221*TF+.1713*TA+.1*TC+.0713*TH+.0659*TFL+.0136*TCI

TKD=-.3874*TK+.0198*TOUT+.1203*TD+.0475*TH+.1203*TC+.0659*TFL+.0136*TCI

TAD=-.1909*TA+.0506*TOUT+.0101*TE+.0214*TF+.014*TH+.0153*TB+.0659*TFL
&+.0136*TCI+QINA

TBD=-.1905*TB+.0527*TOUT+.02*TA+.02*TI+.0183*TH+.0659*TFL+.0136*TCI
&+QINB

TCD=-.1907*TC+.014*TE+.0153*TF+.011*TH+.0278*TK+.0431*TOUT+.0659*TFL
&+.0136*TCI+QINC

TDD=-.1900*TD+.029*TK+.0197*TI+.0136*TH+.0482*TOUT+.0659*TFL
&+.0136*TCI+QIND

TID=-.2443*TI+.0429*TB+.044*TH+.0428*TD+.0351*TOUT+.0659*TFL+.0136*TCI
&+QINI

TCID=-.2118*TCI+.007215*TA+.00553*TB+.002583*TI+.00562*TD
&+.001353*TK+.001968*TH+.000902*TF+.000451*TE+.005863*TC+.1803*TOUT

THD=-.3771*TH+.05143*TA+.05143*TB+.0327*TF+.0327*TC+.0327*TK+.0389*TD
&+.0577*TI+.0659*TFL+.0136*TCI+QINH

TCI=ENTGRL(TCID,TC100)

DIGLSIM (continued)

TH=ENTGRL(THD,THOO)
TB=ENTGRL(TBD,TBOO)
TI=ENTGRL(TID,TIOO)
TF=ENTGRL(TFD,TFOO)
TA=ENTGRL(TAD,TAOO)
TE=ENTGRL(TED,TEOO)
TD=ENTGRL(TDD,TDOO)
TK=ENTGRL(TKD,TKOO)
TC=ENTGRL(TCD,TCOO)

CALL THERM2

QINB=SWB*HEAT*.8416
QINI=SWI*HEAT*1.8035
QINA=SWA*HEAT*.6455
QIND=SWD*HEAT*.8293
QINC=SWC*HEAT*.7945

QTA=ENTGRL(QINA,0.0)
QTB=ENTGRL(QINB,0.0)
QTI=ENTGRL(QINI,0.0)
QTC=ENTGRL(QINC,0.0)
QTD=ENTGRL(QIND,0.0)
QTDD=QINB+QINA+QINC+QIND+QINI
QT=ENTGRL(QTDD,0.0)

HEAT=18
TOUT=TFL=TEMP2
TEMP2=20
TCIOO=TBOO=TIOO=TFOO=TAOO=TEOO=TDOO=TKOO=
&THOO=TEMP1

TCOO=50
TEMP1=75
IF(KERR .NE. 0) GO TO 9
ITZ=ITZ+1
IF(ITZ .LE. ITMAX) GO TO 5
KPZ=0

*** OUTPUT REGION *** LINES 1200-1500

CALL OUTPUT(QT,"TOTAL HEAT")
CALL OUTPUT(TC,"TC")
CALL OUTPUT(TA,"TA")
JJZ=JJZ+2

IF(ICZ .EQ. 0)GO TO 6
ICZ=0
ITMAX=NITZ

6 IF(T .LT. (TMAX+.00001))GO TO 68
IF(TKPLOT .NE. 0)GO TO 7

WRITE(ZOUT,12)

12 FORMAT(" WANT TO INCREASE TMAX")

DIGISIM (continued)

```
      READ(ZOUT,14)IPQ
14  FORMAT(V)
      IF(IPQ.NE."YES") GO TO 7
      WRITE(ZOUT,13)
13  FORMAT(" TMAX=")
      READ(ZOUT,14)TMAX
      GO TO 6
68  KTZ=KTZ+1
      T=KTZ
      T=T*DELT+T0
      GO TO 4
9   WRITE(ZOUT,10)
10  FORMAT(5X," INTEGRATION ASSIGNMENT ERROR")
      STOP
7   ISZZ=2
****  TERMINAL REGION  ***  LINES 1700-1850
*
*
*
*
*
****
      IF(TKPLOT.EQ.0)CALL RANGE
      IF(TKPLOT.EQ.0) CALL PLOT2
      IF(TKPLOT.NE.0) CALL TEKPLT
      STOP
      END
      FUNCTION ENTGRL(X,Y0)
      COMMON OUT,XVZ(6,402),ICNT,NVAR,T,TO,TMAX,DELT,HDELT,IC,NIC,IT,NIT,
&ITMAX,KERR,IA(100),Y1(100),Y2(100),X1(100),X2(100),N,KP,NSAMPL,KT
      N=N+1
      IF(IC.EQ.0) GO TO 20
      IF(IT.NE.1) GO TO 10
      IF(IA(N).NE.0) GO TO 50
      IA(N)=1
      GO TO 15
10  IF(IA(N).EQ.0) GO TO 50
15  Y2N=Y0
      GO TO 40
20  IF(IA(N).EQ.0) GO TO 50
      IF(IT.NE.1) GO TO 30
*    PREDICTOR
      Y1N=Y2(N)
      X1N=X2(N)
      Y1(N)=Y1N
      X1(N)=X1N
      Y2N=Y1N+X1N*DELT
*    CORRECTOR
30  Y2N=Y1(N)+(X1(N)+X)*HDELT
40  Y2(N)=Y2N
```

DIGISIM (continued)

```
X2(N)=X
ENTGRL=Y2N
RETURN
50 KERR=1
RETURN
END
FUNCTION STEP(T1)
COMMON OUT,XVZ(6,402),ICNT,NVAR,T
Y=1.
IF(T.LT.T1) Y=0.
STEP=Y
RETURN
END
FUNCTION PULSE(T1,T2)
COMMON OUT,XVZ(6,402),ICNT,NVAR,T
Y=0.
IF(T.GE.T1.AND.T.LE.T2) Y=1.
PULSE=Y
RETURN
END
FUNCTION SQUARE(T1,T2,T3)
COMMON OUT,XVZ(6,402),ICNT,NVAR,T,TO,TMAX,DELT,HDELT
Y=0.
TM=AMOD(T-T1,T3)
IF(TM.GE.0.AND.TM.LE.T2) Y=1.
SQUARE=Y
RETURN
END
FUNCTION FRSTOR(X,A,B,C,D)
COMMON /FRST/ SZ,Z
IF(C.EQ.ZERO) GO TO 1
SZ=X-D*Z/C
Z=ENTGRL(SZ,0.)
FRSTOR=(A*SZ+B*Z)/C
RETURN
1 PRINT 2
2 FORMAT(5X,"FIRST ORDER TRANSFER FUNCTION OUT OF ORDER")
STOP
END
FUNCTION SCNDOR(X,A,B,C,D,E,F)
COMMON /SCND/ S2Z,SZ,Z
IF(D.EQ.ZERO) GO TO 1
S2Z=X-(E*SZ+F*Z)/D
SZ=ENTGRL(S2Z,0.)
Z=ENTGRL(SZ,0.)
SCNDOR=(A*S2Z+B*SZ+C*Z)/D
RETURN
1 PRINT 2
2 FORMAT(5X,"SECOND ORDER TRANSFER FUNCTION OUT OF ORDER")
STOP
```

NOT-EXISTING PAGE
BY MISNUMBERING.

NOT-EXISTING PAGE
BY MISNUMBERING.

NOT-EXISTING PAGE
BY MISNUMBERING.

DIGISIM (continued)

```
END
SUBROUTINE OUTPUT(X,A1)
CHARACTER A1,A(25),HEAD*80
COMMON OUT,XVZ(6,402),ICNT,NVAR,T,TO,TMAX,DELT,HDELT,IC,NIC,IT,NIT,
&ITMAX,KERR,IA(100),Y1(100),Y2(100),X1(100),X2(100),N,KP,NSAMPL,KT
COMMON /OUTP2/II,JJ,A,ISZZ,AMXZ(25),AMNZ(25),HEAD,NPLOT,TKPLOT,PPLANE
COMMON /OUTP1/ V(25),IP,KASE,KS,TSAB,ICN
*
  IF(T .NE. TO) GO TO 10
  KS = 0
  IF(JJ .GT. 0) GO TO 40
  II=II+1
  A(II)=A1
  ISZZ=1
  NVAR = MIN(II,6,NPLOT)
  GO TO 40
10 IF(KP .NE. 0) GO TO 40
  IF(KS .EQ. 0) GO TO 60
20 KS=KS+1
  IF(KS .GE. NSAMPL) KS = 0
40 KP=KP+1
  IF(KS .NE. 0) RETURN
  TSAB = T
  V(KP) = X
  RETURN
* PRINT LAST TIME PERIOD
60 CONTINUE
  ICNT=ICNT+1
  DO 120 I=1,NVAR
120 XVZ(I,ICNT) = V(I)
  IF(TKPLOT .NE. 0) GO TO 20
  IF(ICN .EQ. 0) WRITE(OUT,900)
  IF(ICN .EQ. 0) PRINT,HEAD
  IF(ICN .LT. 51) GO TO 70
  ICN = 0
  WRITE(OUT,915)
  WRITE(OUT,900)
70 CONTINUE
  KT1=MIN(5,II)
  IF(JJ .EQ. 2) WRITE(OUT,910)(A(I),I=1,KT1)
  IF(II .GT. 5) WRITE(OUT,905)
  WRITE(OUT,920)ICNT,TSAB,(V(I),I=1,KT1)
  ICN=ICN+1
  IF(II .LE. 5) GO TO 100
*
  ICN = ICN + 1
80 KT1=KT1 + 1
  KT2=MIN(KT1+4,II)
  IF(JJ .EQ. 2) WRITE(OUT,930)(A(I),I=KT1,KT2)
  WRITE(OUT,940)(V(I),I=KT1,KT2)
```

DIGISIM (continued)

```

      ICN=ICN+1
      KT1 = KT2
      IF(II .GT. KT2) GO TO 80
100  JJ=4
      CALL RANGE
      GO TO 20
900  FORMAT(1H1,////////)
905  FORMAT(1H )
910  FORMAT(/11X,"TIME",3X,5(2X,A8,2X))
915  FORMAT(////)
920  FORMAT(1H ,I3,1X,6G12.5)
930  FORMAT(/,18X,5(2X,A8,2X))
940  FORMAT(1H ,15X,5G12.5)
      END
      SUBROUTINE PLOT2

```

```

      N= NUMBER OF POINTS
      NVAR= NUMBER OF PLOTS

```

```

      V= NVAR X N MATRIX; EACH COLUMN CORRESPONDS TO A VECTOR
        COMPOSED OF POINTS TO BE PLOTTED

```

```

      MAX NVAR = 6
      MAX N = 400 WITHOUT CHANGING DIMENSIONS

```

```

      COMMON OUT,V(6,402),N,NVAR
      COMMON /OUTP2/II,JJ,A2,ISZZ,AMXZ(25),AMNZ(25),HEAD,NPLOT,TKPLOT,
&PPLANE

```

```

      DIMENSION SF(7),IS(7),H(7),IC(7)
      CHARACTER BLANK,DOT,STR,MINS,HEAD*80
      CHARACTER X(8),LINE(120),A2(25)
      BLANK=" "
      DOT="I"
      STR="+"
      MINS="-"
      X(1)="+"
      X(2)="*"
      DO 9999 I=1,7,2
9999X(I)=X(1)
      DO 9998 I=2,8,2
9998X(I)=X(2)

```


DIGISIM (continued)

```
DO 1 I=1,NVAR
H(I)=0
SF(I)=0.0
IS(I)=I
1 IC(I)=1
*
DO 6 I=1,NVAR
V(I,1)=V(I,1)+1.E-6
DO 6 J=1,N
IF(V(I,J) )3,3,4
3 IC(I)=2
4 IF(ABS(V(I,J))-H(I))6,6,5
5 H(I)=ABS(V(I,J))
6 CONTINUE
*
*
* GENERATE SCALE FACTORS
*
DO 7 I=1,NVAR
A=60/(NVAR*IC(I))
SF(I)=A/(H(I)+.1*H(I))
A=(30 *(IC(I)-1)+60 *(I-1))/NVAR
DO 7 J=1,N
7 V(I,J)=SF(I)*V(I,J)+A
*
*
* WRITE HEADING
*
WRITE(OUT,500)NVAR
IF(NVAR-4)8,8,9
8 NS=1
NQ=1
NP=NVAR
GO TO 10
9 NS=2
NQ=1
NP=4
10 DO 11 K=1,NS
WRITE(OUT,501)(A2(I),I=NQ,NP)
WRITE(OUT,502)(SF(I),I=NQ,NP)
NQ=5
11 NP=NVAR
WRITE(OUT,505)
PRINT,HEAD
WRITE(OUT,503)
*
*
* MAKE AXIS
```

DIGISIM (continued)

```
*
*
LLNTH = 61
K=0
DO 12 I=1,LLNTH
12 LINE(I)=MINS
WRITE(OUT,504)K,(LINE(I),I=1,LLNTH)
*
*
BLANK THE LINE
*
*
DO 13 I=1,LLNTH
13 LINE(I)=BLANK
*
*
PLOT THE VARIABLES
*
*
DO 15 J=1,N
LINE(1)=DOT
DO 14 I=1,NVAR
JK=(60 *I)/NVAR +1
JP=JK-(30 *(IC(I)-1))/NVAR
JL=V(I,J)+2.5
LINE(JK)=DOT
LINE(JP)=DOT
14 LINE(JL)=X(1)
WRITE(OUT,504)J,(LINE(K),K=1,LLNTH)
DO 15 K=1,LLNTH
15 LINE(K)=BLANK
*
*
RETURN
*
*
500 FORMAT( ///5X,"SYSTEM PLOT IN"/5X,I2,5X,"VARIABLES"/)
501 FORMAT(5X,"VARIABLES;",5X,4(A8,6X))
502 FORMAT(3X,"SCALE FACTORS",4(G14.5))
503 FORMAT(///1X,6(10X,"1"))
504 FORMAT( 1X,I4,6X,121A1)
505 FORMAT(//)
END
FUNCTION DELAY(C,TLAG)
CHARACTER A(25),HEAD*80
COMMON OUT,XVZ(6,402),ICNT,NVAR,T,TO,TMAX,DELT,HDELT,IC,NIC,
&IT,NIT,ITMAX,KERR,IA(100),Y1(100),Y4(100),X1(100),X4(100),N,KP,
&NSAMPL,KT
```

DIGISIM (continued)

```
COMMON /OUTP2/ II,JJ,A,ISZZ,AMX(25),AMN(25),HEAD,NPLOT,TKPLOT,
&PPPLANE
COMMON /OUTP1/ V(25),IP,KASE,KS,TSAM,ICN
COMMON /HOLD/ IADCNT,ITQA,JCNT,IQCNT,HLD(500)
IF(IADCNT.GE.1) GO TO 102
IADCNT=1
ITQA=1
JCNT=0
IQCNT=0
102 IF(T.GT.TLAG) GO TO 103
DELAY=HLD(ITQA)
HLD(ITQA)=C
IQCNT=IQCNT+1
GO TO 104
103 IF(JCNT.GE.1) GO TO 106
ITQA=1
JCNT=1
GO TO 107
106 IF(ITQA.LT.IQCNT) GO TO 107
JCNT=0
107 DELAY=HLD(ITQA)
HLD(ITQA)=C
104 ITQA=ITQA+1
RETURN
END
FUNCTION DIGITAL(U)
COMMON OUT,XVZ(6,402),ICNT,NVAR,T,TO,TMAX,DELT,HDELT,IC,NIC,IT,NIT,
&ITMAX,KERR,IA(100),Y1PZ(100),Y4PZ(100),X1PZ(100),X4PZ(100),N,KP,,KT
&NSAMPL
COMMON/DIGITA/IDIGL,ZA(4),ZB(4),TSAM,ZK,NCNT
SAVE Y,Y1,Y2,U1,U2,U3,Y3
*
*
IF(IDIGL .GT. 0) GO TO 100
IDIGL=1
NCNT=0
U1=0.
U2=0.
Y1=0.
Y2=0.
U3=0.
Y3=0.
*
* ADJUST DELT TO BE INTEGRAL MULTIPLE OF TSAM
*
IF(DELT .GT. TSAM/10.) GO TO 50
PP=1.
20 RR=TSAM/PP/DELT
JJ=RR
IF((RR-FLOAT(JJ)) .LT. .001) GO TO 70
```

DIGISIM (continued)

```

      PP=PP+1.
      IF(PP .LE. 3.) GO TO 20
      PP=1.
30    XDEL=TSAM/20./PP
      IF(XDEL .LT. DELT) GO TO 60
      PP=PP+1.
      GO TO 30
50    XDEL=TSAM/10.
60    NSAMPL=NSAMPL*DELT/XDEL
      DELT=XDEL
      GO TO 100
70    DELT=DELT/PP
      NSAMPL=NSAMPL*PP
100   CONTINUE
*
*   HERE FOR NORMAL OPERATION
*
      IF(ABS(KT*DELT - NCNT*TSAM) .GT. 1.E-6 ) GO TO 300
      IF(IT .EQ. ITMAX) NCNT=NCNT+1
      Y=ZK*(ZA(1)*U+ZA(2)*U1+ZA(3)*U2+ZA(4)*U3)
      Y=(Y-ZB(2)*Y1-ZB(3)*Y2-ZB(4)*Y3)/ZB(1)
      IF(IT .LT. ITMAX) GO TO 300
      U2=U1
      U1=U
      Y2=Y1
      Y1=Y
*
300   DIGTAL=Y
      RETURN
      END
      SUBROUTINE RANGE
      CHARACTER A(25),HEAD*80
      COMMON OUT,XVZ(6,402),ICNT
      COMMON /OUTP2/ II,JJ,A,IS,AMX(25),AMN(25),HEAD,NPLOT,TKPLOT,
&PPLANE
      COMMON /OUTP1/ V(25),IP,KASE,KS,TSAB,ICN
*
      IF(ICNT .GT. 1) GO TO 20
      DO 10 I=1,25
      AMX(I) = -1.E+6
10    AMN(I) = 1.E6
20    GO TO (30,80),IS
30    DO 50 I=1,II
      AMX(I) = AMAX1(AMX(I),V(I))
50    AMN(I) = AMIN1(AMN(I),V(I))
70    RETURN
*
80    WRITE(OUT,900)
      DO 100 I=1,II
100   WRITE(OUT,910)I,A(I),AMX(I),AMN(I)

```


DIGISIM (continued)

```
GO TO 70
900 FORMAT(1H0,"    MAX AND MIN VALUES")
910 FORMAT(1H ,I2,3X,A8," MAX =",G12.5,"  MIN =",G12.5)
END
REAL FUNCTION LIMIT(U1,X,U2)
LIMIT = X
IF(X .GT. U2) LIMIT=U2
IF(X .LT. U1) LIMIT=U1
RETURN
END
SUBROUTINE TEKPLT
```

```
*
* THIS SUBROUTINE PREPARES A FILE FOR PLOTTING
* ON THE TEKTRONICS GRAPHIC TERMINALS.
* YOU MUST OPEN A FILE IN YOUR CATALOG
*
*
```

```
CHARACTER A(25),HEAD*80
COMMON OUT,X(6,402),ICNT,NVAR,T,TO,TMAX,DELT,HDELT,IC,NIC,IT,NIT,
&ITMAX,KERR,IA(100),Y1PZ(100),Y4PZ(100),X1PZ(100),X4PZ(100),N,KP,,KT
&NSAMPL
COMMON /OUTP2/ II,JJ,A,IS,AX(25),AZ(25),HEAD,NPLOT,TKPLOT,PPLANE
```

```
*
*
IF(PPLANE .NE. 0)GO TO 70
DO 50 I=1,NVAR
K=0
T=TO
DO 20 J=1,ICNT
WRITE(3,500)T,X(I,J)
K=K + 1
T = K*NSAMPL
T = T*DELT + TO
20 CONTINUE
WRITE(3,510)
50 CONTINUE
RETURN
70 DO 80 J=1,ICNT
80 WRITE(3,500)X(1,J),X(2,J)
WRITE(3,510)
RETURN
500 FORMAT(E12.4,E12.4)
510 FORMAT(" 1.E37 , 1.E37 ")
END
REAL FUNCTION PTRAIN(T1)
COMMON ZOUT,XVZ(6,402),ICNT,NVAR,T,TO,TMAX
SAVE P,J
DATA J/0/
PTRAIN=0.
P=J
```

DIGISIM (continued)

```
P=P*T1
IF(T.GT.P)J=J+1
IF(ABS(T-P).LT..0001)PTRAIN=1.
RETURN
END
SUBROUTINE THERM1
COMMON/TEMP1/SWA,SWB,SWC,SWD,SWH,SWI
COMMON/TEMP2/TA,TB,TC,TD,TH,TI,TOUT
REF=75
LOW=35
A=.5*(REF-TOUT)+TOUT
IF(A.GT.LOW) GO TO 70
TFLO=LOW
GO TO 71
70 IF(A.GT.REF) GO TO 75
TFLO=A
GO TO 71
75 TFLO=REF
71 IF(T.GT.240) GO TO 1
GO TO 3
1 IF(T.GT.720) GO TO 2
RC=RD=RH=TFLO
RA=RI=REF
RB=LOW
GO TO 4
2 IF(T.GT.1200) GO TO 3
RH=TFLO
RC=RD=RI=REF
RA=RB=LOW
GO TO 4
3 RA=RB=RH=TFLO
RI=REF
RC=RD=LOW
4 CONTINUE
IF(TH.GT.RH) GO TO 845
SWH=1
GO TO 846
845 SWH=0
846 IF(TB.GT.RB) GO TO 849
SWB=1
GO TO 850
849 SWB=0
850 IF(TI.GT.RI) GO TO 853
SWI=1
GO TO 854
853 SWI=0
854 IF(TA.GT.RA) GO TO 857
SWA=1
GO TO 858
857 SWA=0
```

DIGISIM (continued)

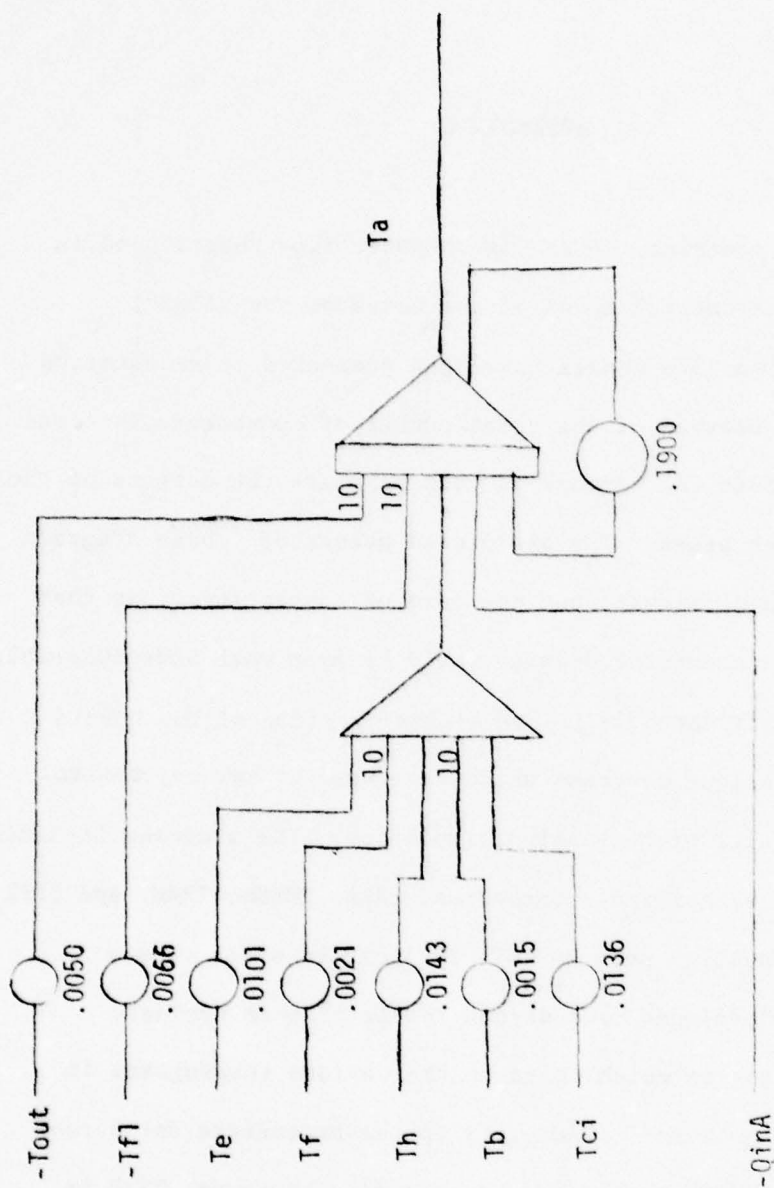
```
858 IF(TD.GT.RD) GO TO 861
    SWD=1
    GO TO 862
861 SWD=0
862 IF(TC.GT.RC) GO TO 865
    SWC=1
    GO TO 866
865 SWC=0
866 CONTINUE
    RETURN
    END
    SUBROUTINE THERM2
    COMMON/TEMP1/SWA,SWB,SWC,SWD,SWH,SWI
    COMMON/TEMP2/TA,TB,TC,TD,TH,TI,TOU

    REF=75
    RA=RB=RD=RH=RI=REF
    RC=50
846 IF(TB.GT.RB) GO TO 849
    SWB=1
    GO TO 850
849 SWB=0
850 IF(TI.GT.RI) GO TO 853
    SWI=1
    GO TO 854
853 SWI=0
854 IF(TA.GT.RA) GO TO 857
    SWA=1
    GO TO 858
857 SWA=0
858 IF(TD.GT.RD) GO TO 861
    SWD=1
    GO TO 862
861 SWD=0
862 IF(TC.GT.RC) GO TO 865
    SWC=1
    GO TO 866
865 SWC=0
866 CONTINUE
    RETURN
    END
```

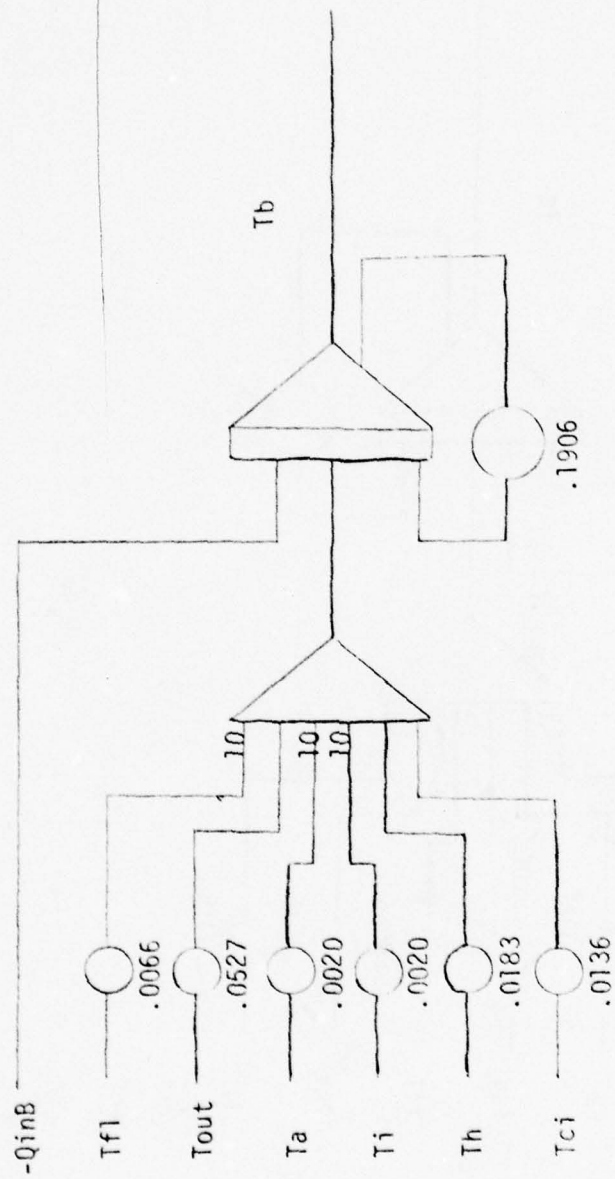
APPENDIX C

This Appendix contains the analog computer flow charts used in programming the mathematical model of the house on the EAI-681 analog computer. The flow charts have been presented in an equation by equation basis, because of the great number of components involved. The input signals into the summers on each page are the outputs of the integrators on other pages. The disjointed nature of these diagrams make interpretation difficult, but the maze of connecting lines that would result from a connected display would be even more undecipherable. Also included in this Appendix is the digital portion of the hybrid simulation. The various programs which were used to set up, control and record the results of the analog simulation. The programs included are HEAT, with its associated subprograms, PARA, THERM, TRAN, and OPT1. The potentiometer setting program POT1 is included also. These programs were all developed as a direct result of this project. HEAT is the main program which calls up the various subprograms in order to run the simulation. PARA sets up the parameters for a run of the simulation, THERM is the heater control subprogram, TRAN is the subprogram which transfers data to and from the analog computer, and OPT1 is the digital controller subprogram.

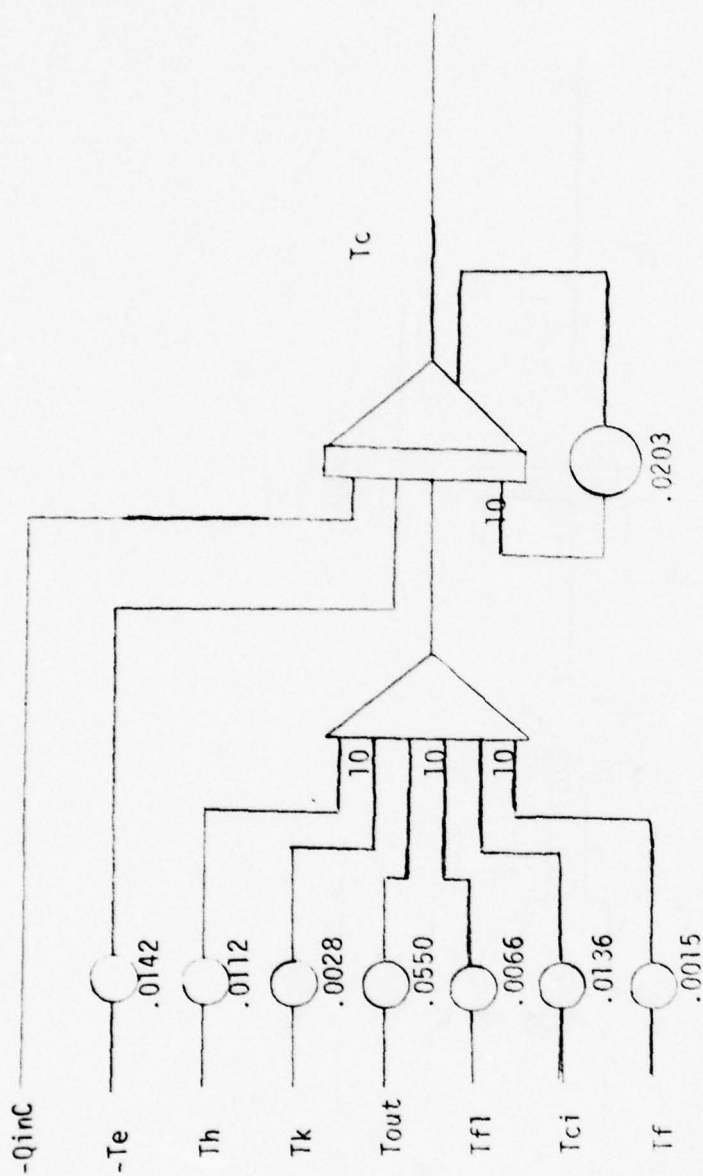
APPENDIX C



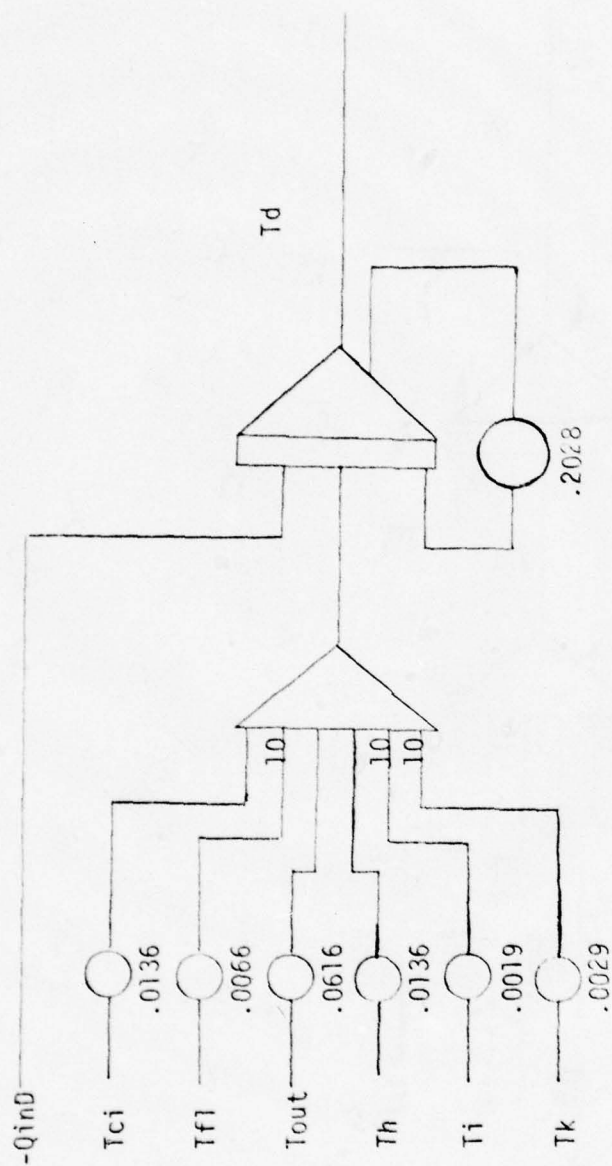
APPENDIX C



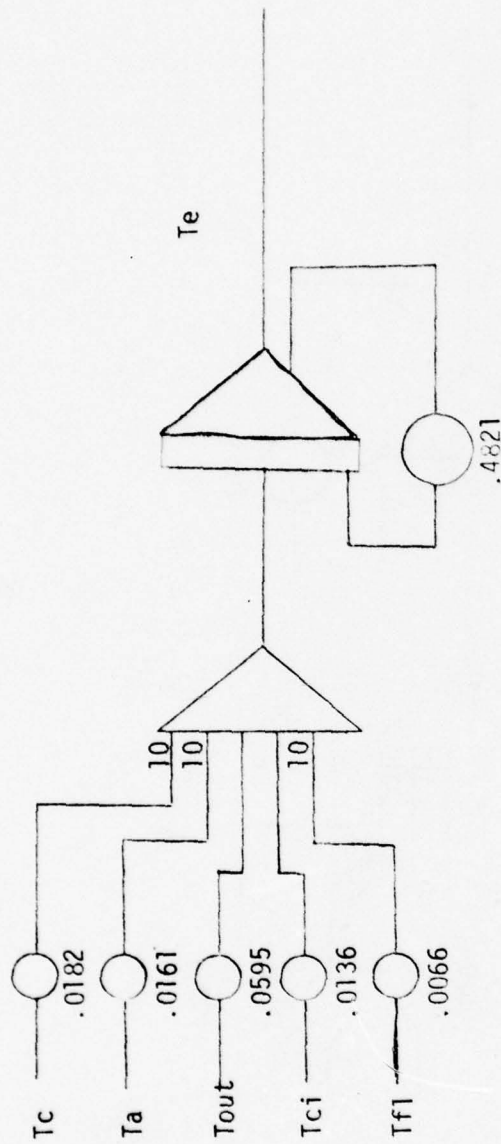
APPENDIX C



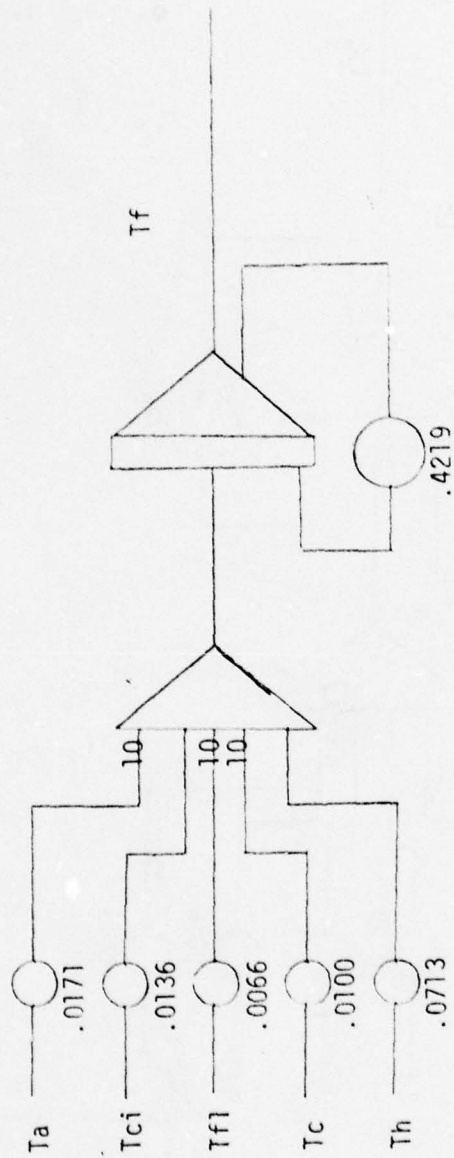
APPENDIX C



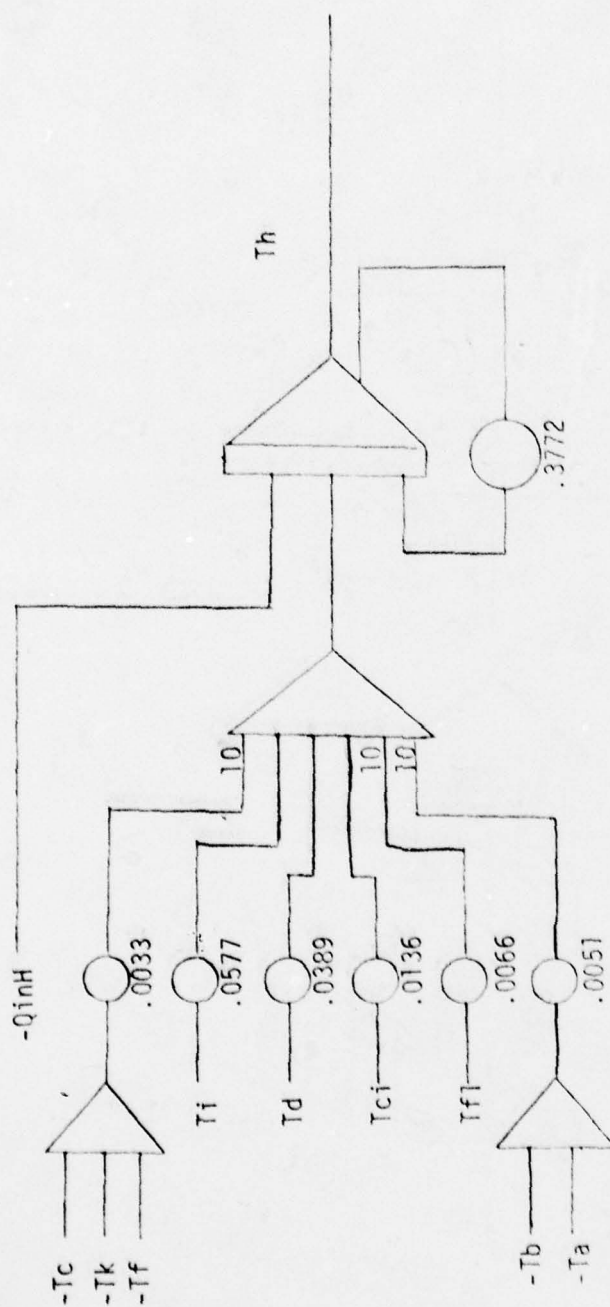
APPENDIX C



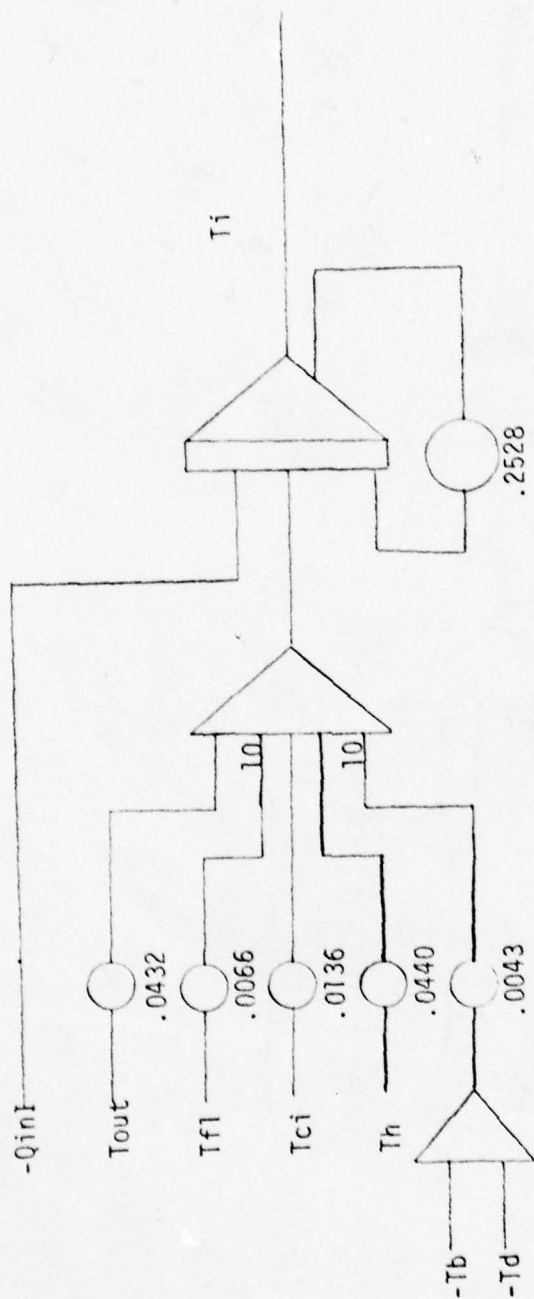
APPENDIX C



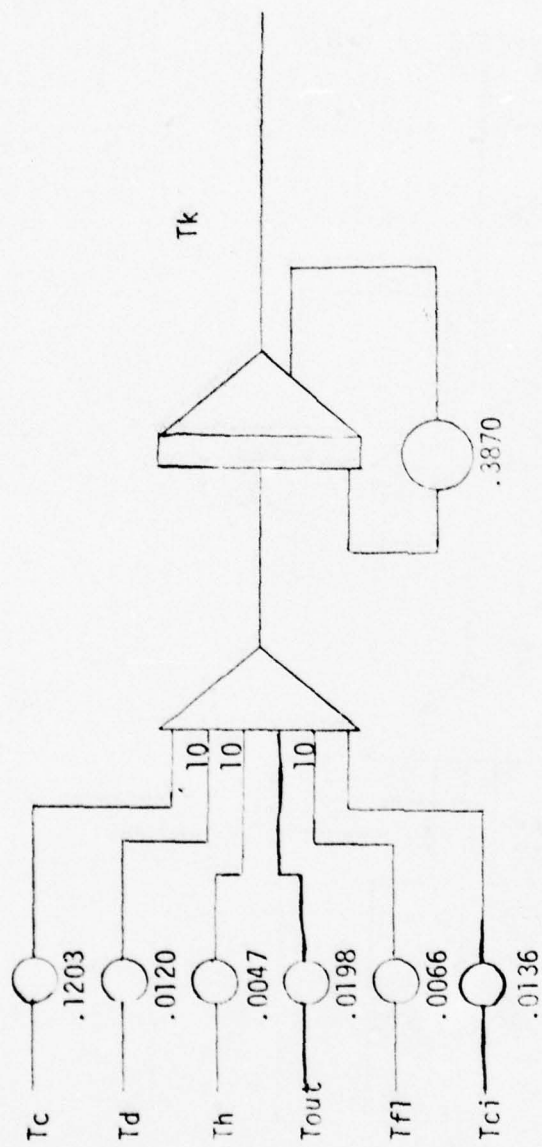
APPENDIX C



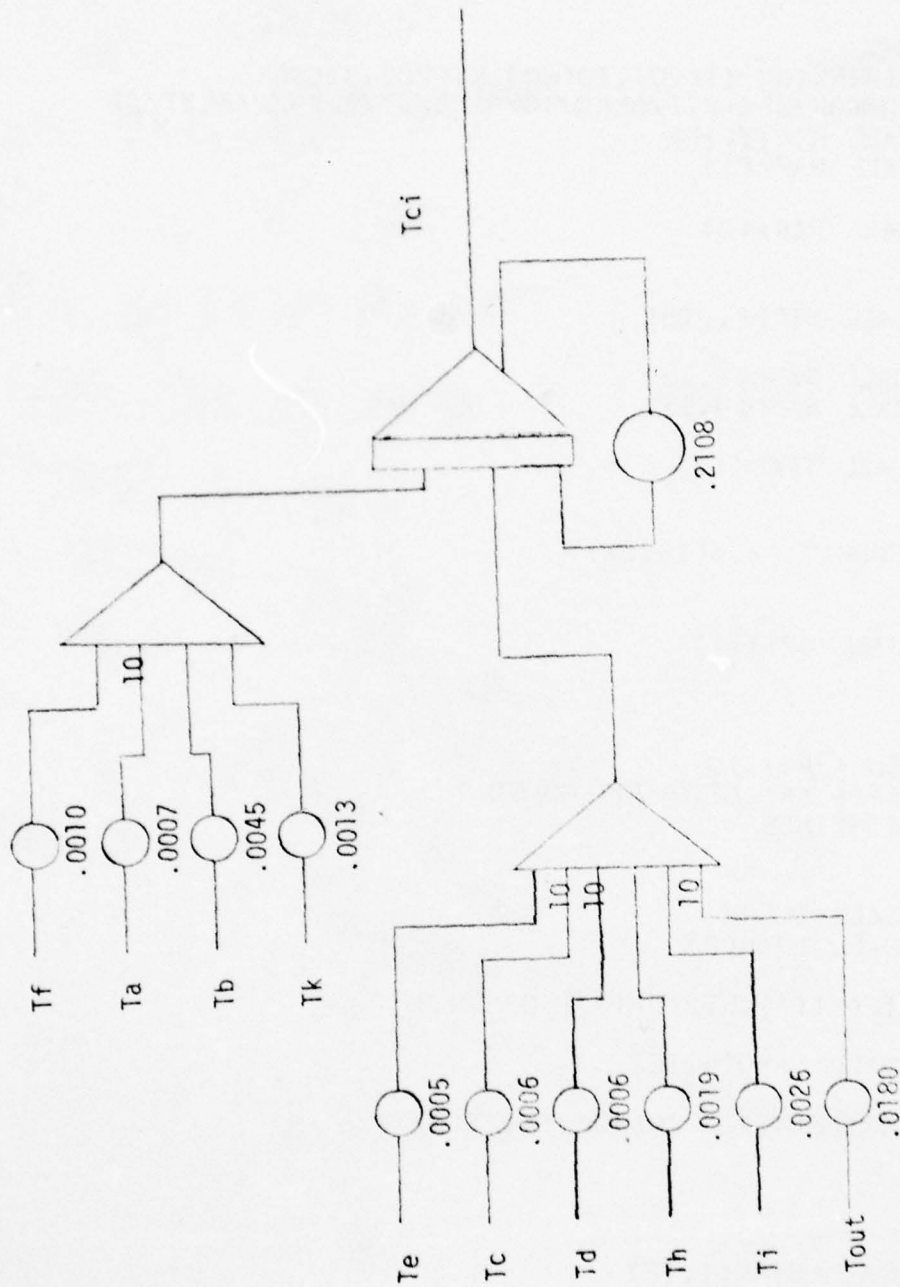
APPENDIX C



APPENDIX C



APPENDIX C



APPENDIX C

```
HEAT
DIMENSION IT(20),IQ(20),KT(20),JT(20)
COMMON/BLK1/IT/BLKQ/IQ/BLKR/JT/BLKP/P/BLKT/KT
5  CALL HINIT(IE)
   CALL PARA(1)

40  CALL PARA(2)

   CALL STCO(1,IE)

   CALL SAMO(6,IE)
   CALL SAMO(4,IE)

8   CALL TRAN(1)

20  FORMAT(1X,6(I5,3X))

   CALL OPTI(1)

   DO 4 M=1,10
4   IF(IT(M).GT.9970) GO TO 7
   CONTINUE

   CALL THERM
   CALL TRAN(2)

   IF(KT(1).LT.KT(10)) GO TO 8

   CALL SAMO(5,IE)

   WRITE(4,20) IQ(10),IT(KT(20))

7   CALL SAMO(1,IE)
   END
```

APPENDIX C

```
SUBROUTINE PARA(K)
  INTEGER OPT(20)
  DIMENSION JT(20),P(5),KT(20),IQ(20),IT(20)
  COMMON/BLKT/KT/BLKQ/IQ/BLKR/JT/BLKO/OPT
  COMMON/BLKP/P/BLKI/IT
  CALL HINIT(IE)

  GO TO (500,1000),K

500  CONTINUE

      WRITE(4,100)
      READ(4,110) LPT
100  FORMAT(32H NORMAL PARAMETERS? YES=1; NO=0)
110  FORMAT(I2)
      IF(LPT.EQ.1) GO TO 170

      WRITE(4,120)
      READ(4,130) JT(12)
120  FORMAT(25H INPUT THERMOSTAT SETTING)
130  FORMAT(I5)

      WRITE(4,140)
      READ(4,130) P(2)
140  FORMAT(25H INPUT HEATER SIZE BTU/HR)

      WRITE(4,160)
      READ(4,130) KT(2)
160  FORMAT(35H INPUT RUN TIME IN SIMULATION-HOURS)
      KT(11)=300*KT(2)

      WRITE(4,165)
      READ(4,130) KT(10)
165  FORMAT(25H INPUT INITIAL CONDITIONS)

      WRITE(4,167)
      READ(4,168) IQ(15)
      READ(4,168) IQ(16)
167  FORMAT(20H INPUT IQ(15),IQ(16))
168  FORMAT(2I5)
      GO TO 300
```


APPENDIX C

PARA (cont.)

170 KT(11)=7500
 KT(10)=7200
 P(2)=200
 JT(12)=JT(8)=JT(4)=JT(10)=JT(2)=JT(3)=7500
 IQ(5)=IQ(6)=2000
 KT(12)=0020

300 CALL SAMO(7,IE)
 CALL SAMO(7,IE)

 CALL SPOT(002,0050,0002,IE)
 CALL SPOT(012,0100,0002,IE)

 CALL SPOT(010,KT(11),0002,IE)

 GO TO 44

1000 CONTINUE

 CALL SPOT(006,JT(20),0002,IE)

44 WRITE(4,44)
 FORMAT(17H ANALOG OPERATING)

 RETURN
 END

APPENDIX C

SUBROUTINE THERM

DIMENSION M(5),IS(10),IT(20),IQ(20),P(5),JT(20)
COMMON/BLK1/IT/BLKQ/IQ/BLKR/JT/BLKP/P

R=.5

M(1)=4

M(2)=10

M(3)=3

M(4)=8

M(5)=2

DO 5 L=1,5

IF(IT(M(L)).GT.JT(M(L))) GO TO 1

IS(M(L))=1

GO TO 5

IS(M(L))=0

CONTINUE

1
5

IQ(4)=IS(4)*P(2)*18.035*R

IQ(3)=IS(10)*P(2)*8.416

IQ(1)=(IS(4)*R+IS(10)+IS(2)+IS(8)+IS(3))*P(2)

IQ(8)=IS(2)*P(2)*7.945

IQ(7)=IS(8)*P(2)*8.293

IQ(2)=IS(3)*P(2)*6.455

RETURN

END

APPENDIX C

SUBROUTINE TRAN(L)
 DIMENSION IT(20),IQ(20),KT(20)
 COMMON/BLK1/IT/BLKQ/IQ/BLKT/KT
 CALL HINIT(IE)

GO TO (1,3),L

1 DO 2 K=1,10
2 CALL CRACS(K-1,IT(K),IE)
 CALL CRACS(012,KT(1),IE)
 CALL CRACS(011,IQ(10),IE)
 GO TO 5
3 DO 4 K=1,8
 CALL LTDAS(K-1,IQ(K),IE)
4 CALL TLDA
5 RETURN
 END

APPENDIX C

SUBROUTINE OPT1(K)
INTEGER OPT(20)
DIMENSION IQ(20),KT(20),JT(20),IO(20)
COMMON/BLKT/KT/BLKR/JT/BLKO/IQ/BLKO/OPT
HI=7500
LOW=3500

7 GO TO (7,5),K
CONTINUE

IF(KT(1).GT.1200) GO TO 8
JT(4)=JT(2)=JT(8)=LOW
JT(10)=JT(4)=HI

8 IF(KT(1).GT.2700) GO TO 1
GO TO 3

1 IF(KT(1).GT.4500) GO TO 2
JT(8)=LOW
JT(3)=JT(2)=JT(4)=JT(10)=HI
GO TO 4

2 CONTINUE

JT(2)=JT(8)=JT(4)=HI
JT(3)=JT(10)=LOW
GO TO 4

3 JT(3)=JT(10)=JT(4)=JT(2)=JT(8)=LOW
JT(3)=JT(10)=JT(4)=HI

4 CONTINUE
GO TO 6

5 JT(4)=JT(10)=JT(3)=JT(8)=JT(2)=7500

6 RETURN

END

APPENDIX C

POT1

IT=0002

```
CALL HINIT(IE)
CALL SAMO(7,IE)
CALL SPOT(057,0051,IT,IE)
K=057
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(023,0033,IT,IE)
K=023
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(016,0033,IT,IE)
K=016
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(020,0043,IT,IE)
K=020
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(063,0104,IT,IE)
K=063
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(061,0022,IT,IE)
K=061
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(064,0143,IT,IE)
K=064
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(062,0015,IT,IE)
K=062
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(032,0136,IT,IE)
K=032
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(060,1900,IT,IE)
K=060
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(072,0066,IT,IE)
K=072
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(069,0050,IT,IE)
K=069
IF(IE.NE.0) WRITE(4,1)IE,K
```

POT1 (cont.)

APPENDIX C

C SET POTS FOR ROOM B

```
CALL SPOT(111,0020,IT,IE)
K=111
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(095,0020,IT,IE)
K=095
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(091,0183,IT,IE)
K=091
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(110,1906,IT,IE)
K=110
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(115,0527,IT,IE)
K=115
IF(IE.NE.O) WRITE(4,1)IE,K
```

C SET POTS FOR ROOM E

```
CALL SPOT(018,0181,IT,IE)
K=018
IF(IE.NE.O)WRITE(4,1)IE,K
CALL SPOT(101,0160,IT,IE)
K=101
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(100,4821,IT,IE)
K=100
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(102,0595,IT,IE)
K=102
IF(IE.NE.O) WRITE(4,1)IE,K
```

C SET POTS FOR ROOM K

```
CALL SPOT(017,1203,IT,IE)
K=017
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(033,0120,IT,IE)
K=033
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(037,0047,IT,IE)
K=037
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(030,3870,IT,IE)
K=030
IF(IE.NE.O) WRITE(4,1)IE,K
CALL SPOT(031,0198,IT,IE)
K=031
IF(IE.NE.O) WRITE(4,1)IE,K
```

APPENDIX C

POT1 (cont.)

```
C SET POTS FOR ROOM F
  CALL SPOT(073,0171,IT,IE)
  K=073
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(013,0100,IT,IE)
  K=013
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(070,4219,IT,IE)
  K=070
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(090,0713,IT,IE)
  K=090
  IF(IE.NE.0) WRITE(4,1)IE,K
C SET POTS FOR ROOM CI
  CALL SPOT(055,0010,IT,IE)
  K=055
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(047,0070,IT,IE)
  K=047
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(112,0045,IT,IE)
  K=112
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(081,0013,IT,IE)
  K=081
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(083,0005,IT,IE)
  K=083
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(027,0060,IT,IE)
  K=027
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(054,0060,IT,IE)
  K=054
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(045,0019,IT,IE)
  K=045
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(097,0026,IT,IE)
  K=097
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(080,0211,IT,IE)
  K=080
  IF(IE.NE.0) WRITE(4,1)IE,K
  CALL SPOT(118,0180,IT,IE)
  K=118
  IF(IE.NE.0) WRITE(4,1)IE,K
```

AD-A058 676

NAVAL ACADEMY ANNAPOLIS MD
MICROPROCESSOR-BASED DIGITAL CONTROL OF ENVIRONMENTAL SYSTEMS.(U)

F/G 9/2

JUN 78 E D KEE

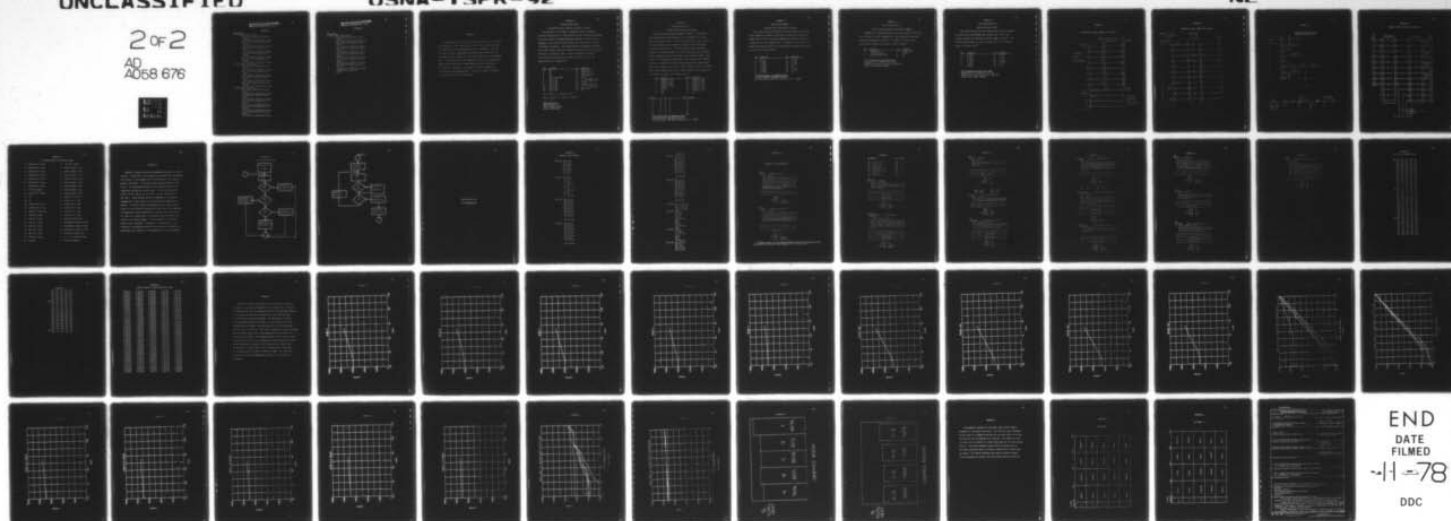
UNCLASSIFIED

USNA-TSPR-92

NL

2 of 2

AD
A058 676



APPENDIX C

POT1 (cont.)

C SET POTS FOR ROOM D

```
CALL SPOT(040,0136,IT,IE)
K=040
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(042,0019,IT,IE)
K=042
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(087,0029,IT,IE)
K=087
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(085,1894,IT,IE)
K=085
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(079,0482,IT,IE)
K=079
IF(IE.NE.0) WRITE(4,1)IE,K
```

C SET POTS FOR ROOM I

```
CALL SPOT(034,0351,IT,IE)
K=034
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(035,0440,IT,IE)
K=035
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(067,0043,IT,IE)
K=067
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(065,2447,IT,IE)
K=065
IF(IE.NE.0) WRITE(4,1)IE,K
```

C SET POTS FOR ROOM H

```
CALL SPOT(046,0033,IT,IE)
K=046
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(071,0577,IT,IE)
K=071
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(077,0389,IT,IE)
K=077
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(105,0052,IT,IE)
K=105
IF(IE.NE.0) WRITE(4,1)IE,K
CALL SPOT(075,3772,IT,IE)
K=075
IF(IE.NE.0) WRITE(4,1)IE,K
```

APPENDIX C

POT1 (cont.)

C SET POTS FOR ROOM C

CALL SPOT(052,0142,IT,IE)

K=052

IF(IE.NE.0) WRITE(4,1)IE,K

CALL SPOT(022,0112,IT,IE)

K=022

IF(IE.NE.0) WRITE(4,1)IE,K

CALL SPOT(053,0028,IT,IE)

K=053

IF(IE.NE.0) WRITE(4,1)IE,K

CALL SPOT(076,0432,IT,IE)

K=076

IF(IE.NE.0) WRITE(4,1)IE,K

CALL SPOT(051,0015,IT,IE)

K=051

IF(IE.NE.0) WRITE(4,1)IE,K

CALL SPOT(050,0192,IT,IE)

K=050

IF(IE.NE.0) WRITE(4,1)IE,K

1 FORMAT(2X,13,2X,13)

END

APPENDIX D

This Appendix contains the specification sheets for the interface unit components. These specification sheets are simplified versions of those put out by the manufacturers of the components. They are not meant for use by someone who is building such a device but for the reader to gain insight into the function and use of the interface device. The connection of the components as well as the interaction between them is more easily understood when a basic knowledge of the components is gained. The schematic for the interface unit is included in this Appendix also. The text of Chapter 4 contains a discussion of the use of this device.

APPENDIX D

SPECIFICATION SHEET 1

ADC-EK8B Monolithic Analog-to-Digital Converter

The ADC-EK8B is a low power, integrating analog-to-digital converter fabricated on a single monolithic chip using CMOS technology. The circuit employs a charge balancing integrator, current switch, comparator, clock counter, data counter, and control logic circuitry to implement conversion. The charge balancing integration technique gives high linearity and noise immunity along with inherent monotonicity resulting in no missing codes. Output data appears in parallel form on latched outputs which are CMOS, low power TTL, or low power Schottky TTL compatible.²

PIN	FUNCTION	PIN	FUNCTION
1	NC	13	Reference
2	NC	14	Analog input
3	NC	15	Amplifier out
4	NC	16	Zero adjust
5	Bit 0 out (MSB)	17	Bias
6	Bit 1 out	18	-5 volt power (in)
7	Bit 2 out	19	+5 volt power (in)
8	Bit 3 out	20	Ground
9	Bit 4 out	21	Start convert (in)
10	Bit 5 out	22	E. O. C. (status)
11	Bit 6 out	23	Data valid
12	Bit 7 out	24	NC

$$R_{in} = FSR/10 \quad A = 10V/10 \quad A = 1M$$

$$R_{ref} = V_{ref}/20 \quad A = +5V/20 \quad A = 250 K$$

²ADC-EK8B Series

Datel Systems, Inc.
1020 Trunpike Street
Canton, Mass 02021

APPENDIX D

SPECIFICATION SHEET 2

CD4051A COS/MOS Analog Multiplexer

The RCA COS/MOS analog multiplexer, CD4051A, is a digitally controlled analog switch having a low "on" impedance and a very low "off" leakage current. Control of analog signals up to fifteen volts peak-to-peak can be achieved by digital signal amplitudes of three to fifteen volts. The multiplexer circuits dissipate extremely low quiescent power over the full V_{DD} - V_{EE} supply voltage range, independent of the logic state of the control signals. When a logic "1" is present at the "inhibit" input terminal all channels are "off". CD4051A is a single 8-channel multiplexer having three binary input signals, A, B, and C, and an "inhibit" input. The three binary input signals select one of eight channels to be turned "on" and connect the input to the common output.³

PIN	FUNCTION	PIN	FUNCTION
1	Channel 4 (in)	9	Control C (in)
2	Channel 6 (in)	10	Control B (in)
3	Common output	11	Control A (in)
4	Channel 7 (in)	12	Channel 3 (in)
5	Channel 5 (in)	13	Channel 0 (in)
6	Inhibit	14	Channel 1 (in)
7	V_{EE}	15	Channel 2 (in)
8	V_{SS}	16	V_{DD}

Inhibit	C	B	A	"ON" Channel
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	-	-	-	none

³ RCA Solid State '74 DATABOOK Series

COS/MOS Digital Integrated Circuits

RCA Solid State, Box 3200, Somerville, N. J. 08876

APPENDIX D

SPECIFICATION SHEET 3

CD4010AE COS/MOS Hex Buffer/Converter

This buffer/converter provides conversion ranges from COS/MOS logic operating at positive three volts to positive fifteen volts supply level to DTL or TTL logic operating at positive three volts to positive six volts supply level. Conversion to logic levels greater than positive six volts is permitted providing V_{CC} (DTL/TTL) is greater than V_{DD} (COS/MOS).⁴

PIN	FUNCTION	PIN	FUNCTION
1	V_{CC}	9	D (in)
2	A (out)	10	D (out)
3	A (in)	11	E (in)
4	B (out)	12	E (out)
5	B (in)	13	NC
6	C (out)	14	F (in)
7	C (in)	15	F (out)
8	Ground	16	V_{DD}

4

RCA Solid State '74 DATABOOK Series

COS/MOS Digital Integrated Circuits

RCA Solid State, Box 3200, Somerville, N. J. 08876

APPENDIX D

SPECIFICATION SHEET 4

High Performance Operational Amplifier μ A741

The μ A741 is a high performance operational amplifier with high open loop gain, internal compensation, high common mode range and exceptional temperature stability. The μ A741 is short-circuit protected and allows for nulling of offset voltages.⁵

PIN	FUNCTION	PIN	FUNCTION
1	Offset null	5	Offset null
2	Inverting input	6	Output
3	Non-inverting input	7	V+
4	V-	8	NC

⁵TTL Integrated Circuits Data Book

Compiled by O. H. Perry III LT, USN

Weapons and Systems Engineering Department,
United States Naval Academy

APPENDIX D

SPECIFICATION SHEET 5

Hex Buffer/Driver N7407

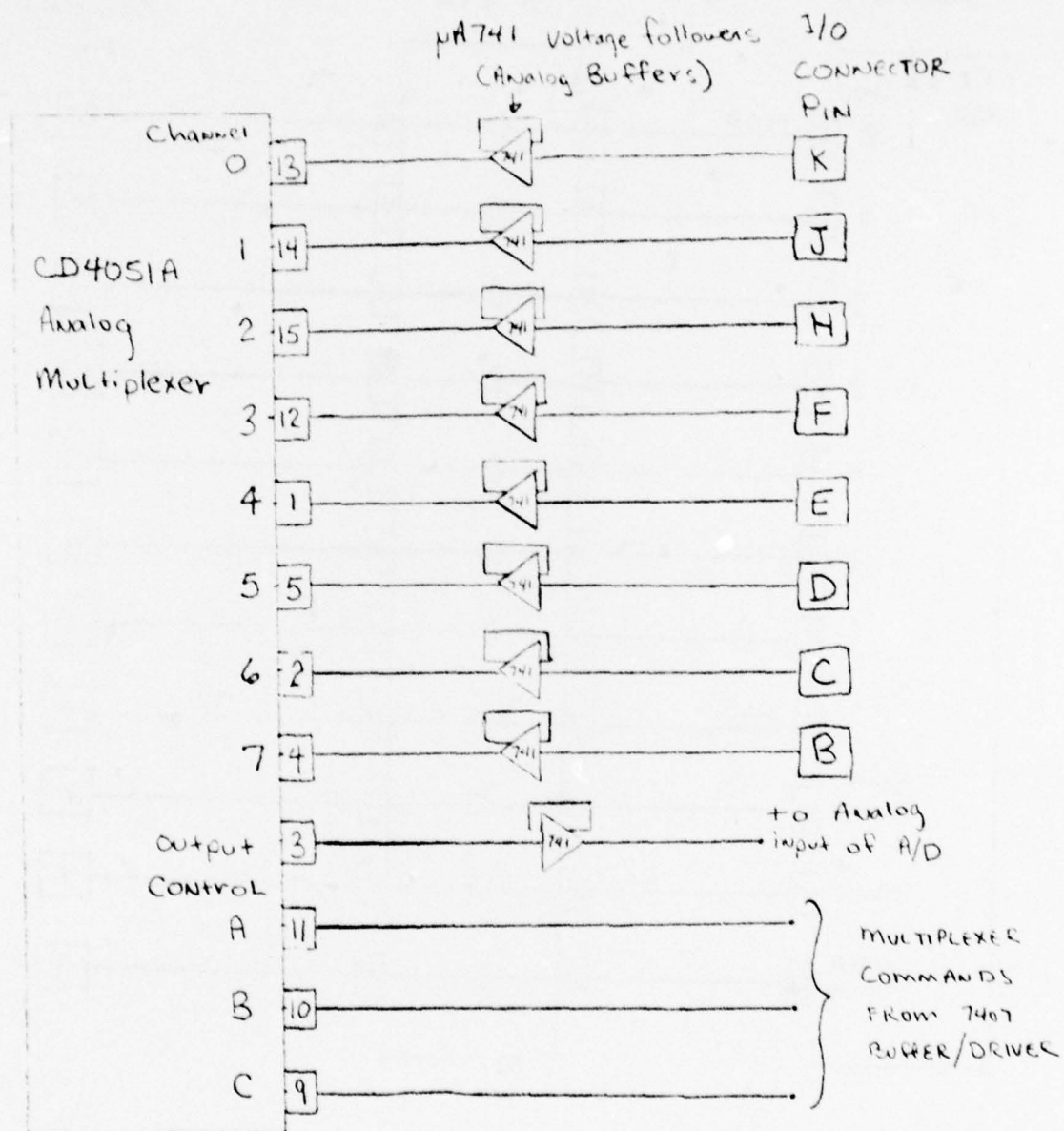
The 54/7407 and 54/7417 Hex buffer/driver features standard TTL inputs with non-inverted high voltage, high current open collector outputs for interface with MOS, lamps, or relays. The 54/7407 maximum output is thirty volts and the 54/7417 maximum output is fifteen volts.⁶

PIN	FUNCTION	PIN	FUNCTION
1	A (in)	8	D(out)
2	A (out)	9	D (in)
3	B (in)	10	E (out)
4	B (out)	11	E (in)
5	C (in)	12	F (out)
6	C(out)	13	F (in)
7	Ground	14	V _{CC}

⁶TTL Integrated Circuits Data Book
Compiled by O. H. Perry III, LT, USN
Weapons and Systems Engineering Department
United States Naval Academy

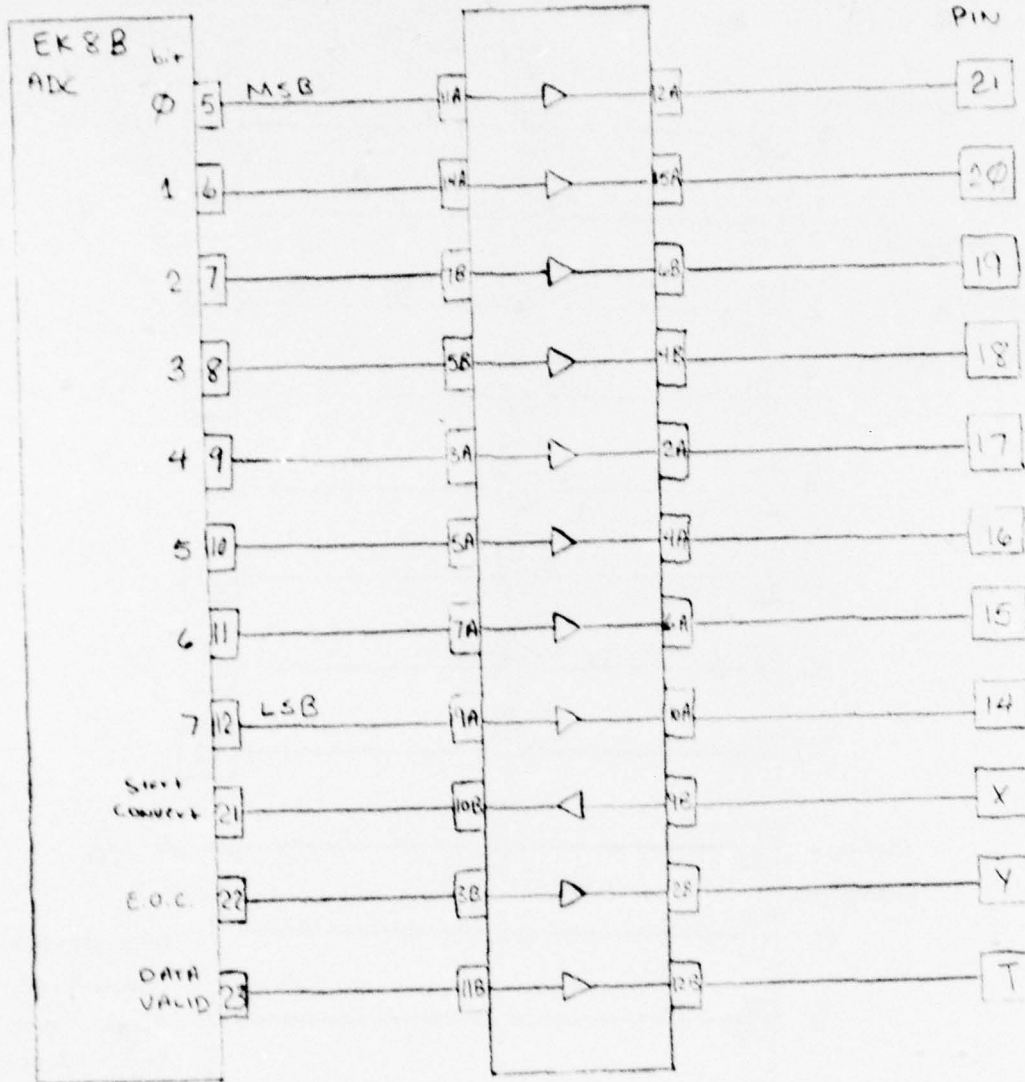
APPENDIX D

MULTIPLEXER SIGNAL CONNECTION SCHEMATIC

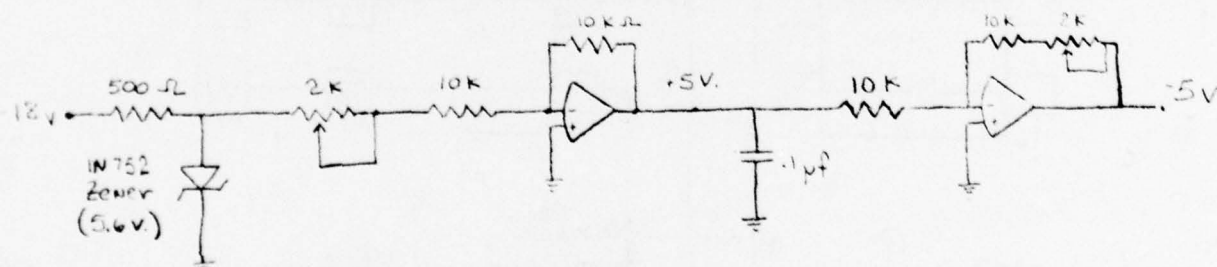
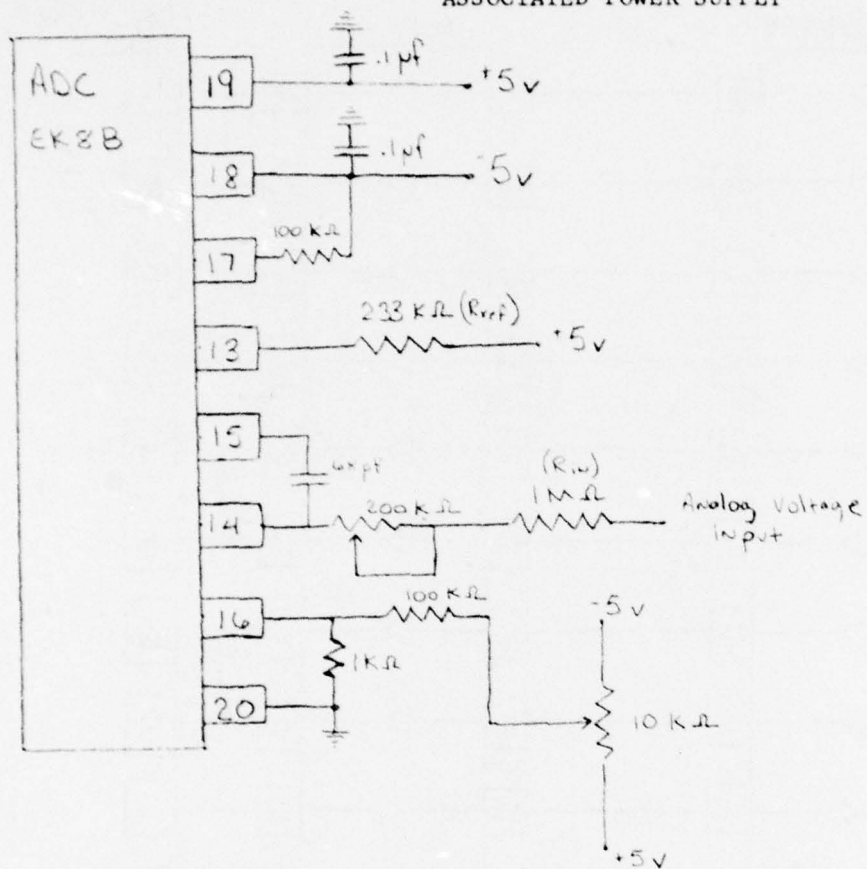


APPENDIX D

EK8B-ADC SIGNAL CONNECTION SCHEMATIC

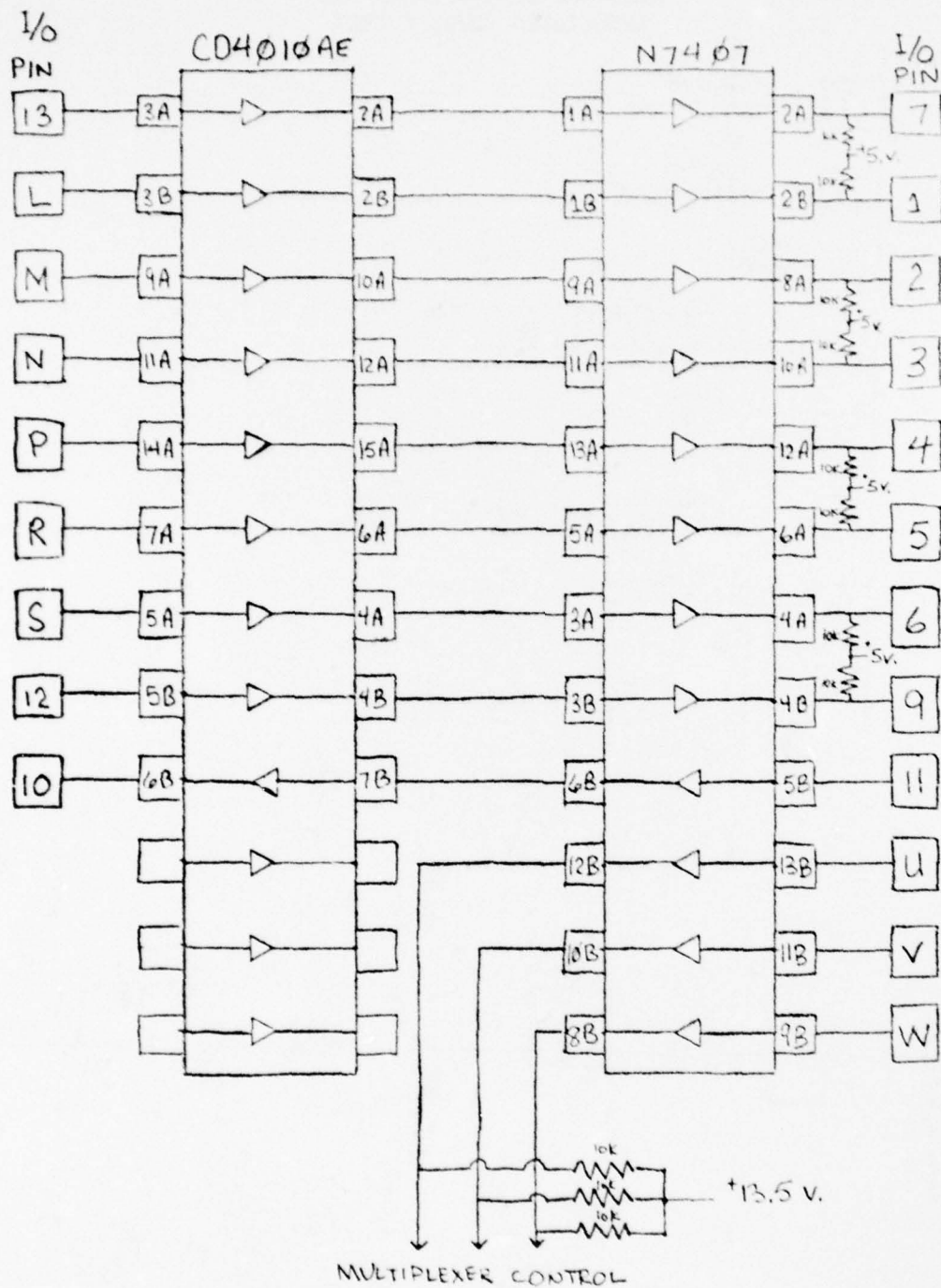
Analog-to-Digital
CONVERTERCD4010 AE
A & BI/O
Connection
PIN

APPENDIX D

EK8B-ADC BIAS NETWORK AND
ASSOCIATED POWER SUPPLY

APPENDIX D

CONTROL SIGNAL CONNECTION SCHEMATIC



APPENDIX D

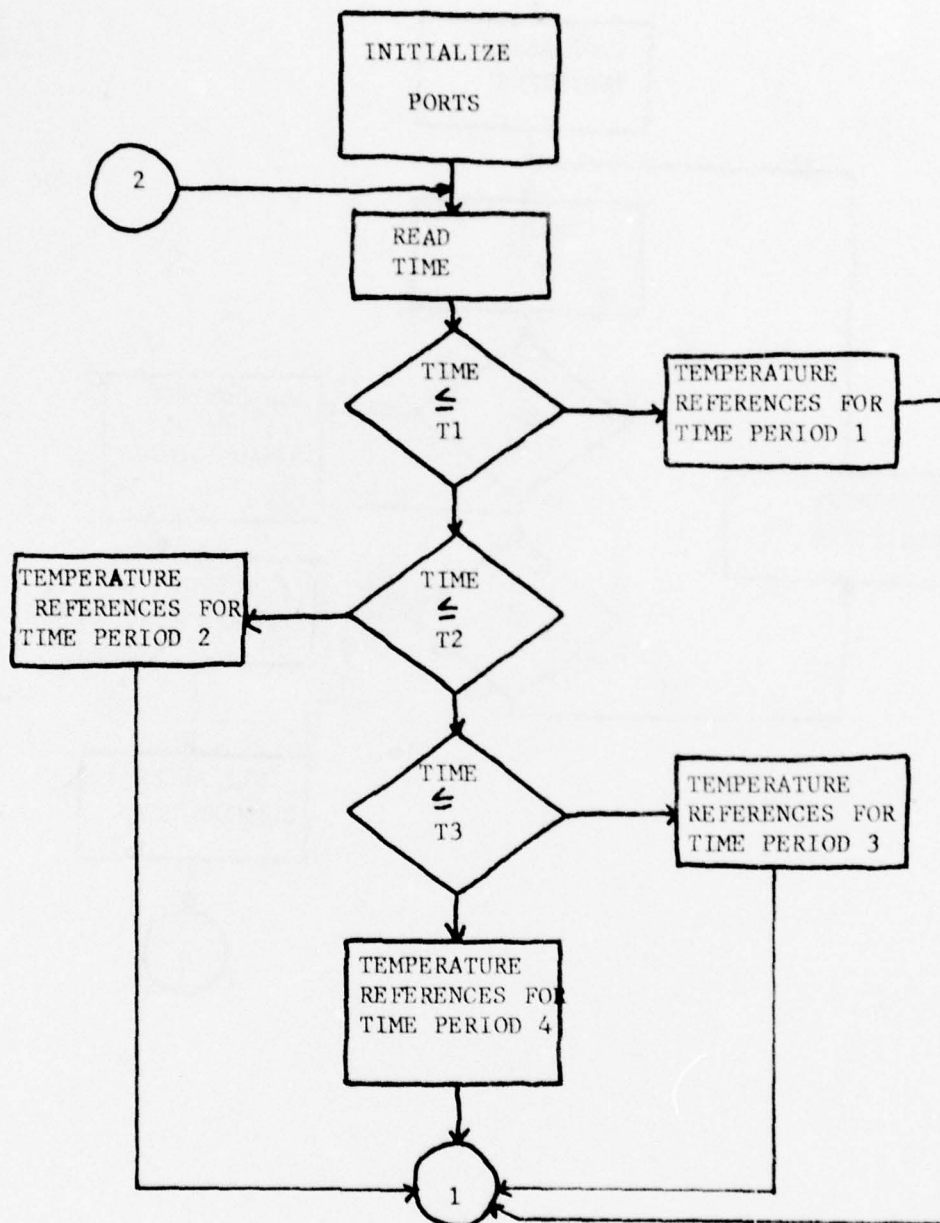
PIN DESIGNATIONS FOR INTERFACE BOARD

1. Control bit 6 (out)	A. -18 volts supply
2. Control bit 5 (out)	B. Analog signal 7 (in)
3. Control bit 4 (out)	C. Analog signal 6 (in)
4. Control bit 3 (out)	D. Analog signal 5 (in)
5. Control bit 2 (out)	E. Analog signal 4 (in)
6. Control bit 1 (out)	F. Analog signal 3 (in)
7. Control bit 0 (out)	H. Analog signal 2 (in)
8. +10 volts supply	J. Analog signal 1 (in)
9. Control bit 7 (out)	K. Analog signal 0 (in)
10. NC	L. Control bit 6 (in)
11. NC	M. Control bit 5 (in)
12. Control bit 7 (in)	N. Control bit 4 (in)
13. Control bit 0 (in)	P. Control bit 3 (in)
14. ADC bit 7 (out) LSB	R. Control bit 2 (in)
15. ADC bit 6 (out)	S. Control bit 1 (in)
16. ADC bit 5 (out)	T. ADC data valid (out)
17. ADC bit 4 (out)	U. Multiplexer control A (in)
18. ADC bit 3 (out)	V. Multiplexer control B (in)
19. ADC bit 2 (out)	W. Multiplexer control C (in)
20. ADC bit 1 (out)	X. ADC start convert (in)
21. ADC bit 0 (out)	Y. E. O. C. status ADC (out)
22. Ground	Z. +18 volts supply

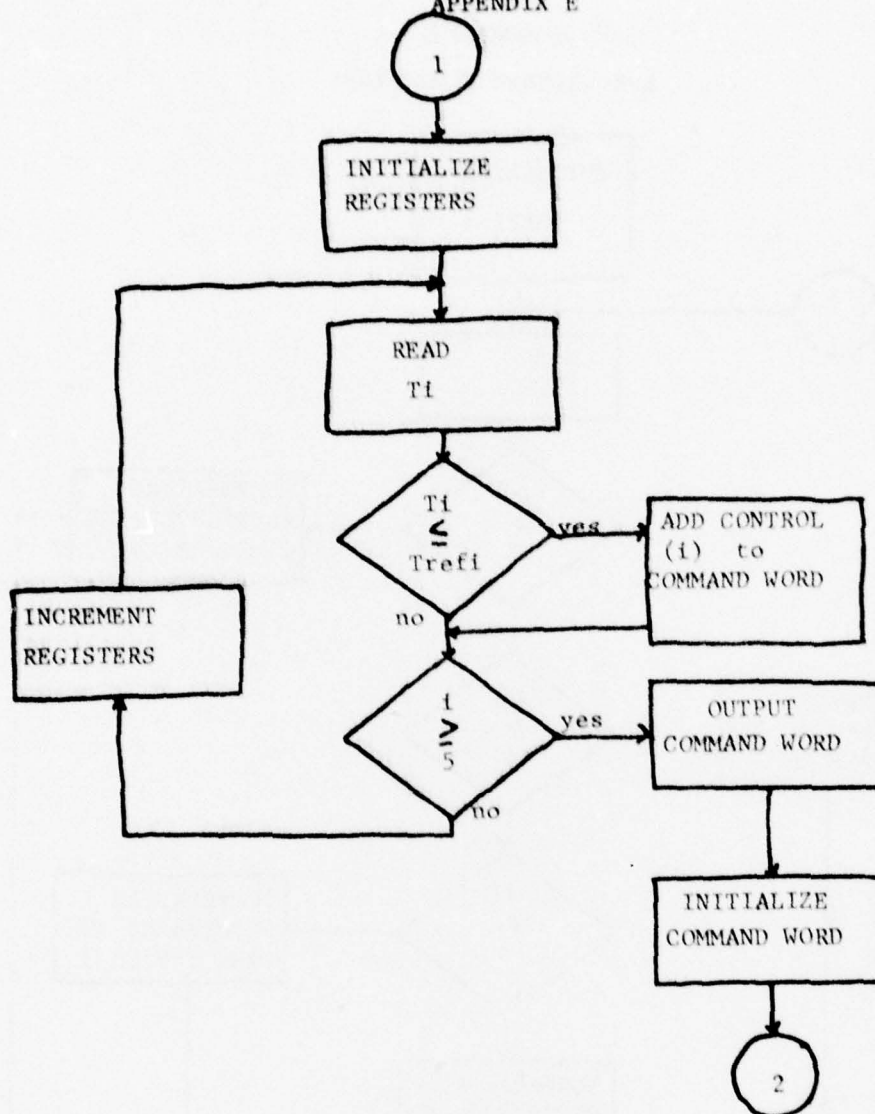
APPENDIX E

Appendix E contains the entire programming sequence for the H-8 computer. A flow chart of the program is presented, then the mnemonic code listing of the program, then the definitions of the various mnemonic code words. The actual machine language program is then listed. The parenthesized numbers in the program represent the temperature settings for various rooms. (1) is the setting for room A, (2) for room B, (3) for room C, (4) for room D, and (5) for room I. These settings would be programmed for the desired temperatures in each room for the time period the temperature is desired. The octal numbers corresponding to the analog voltage levels in the EAI-681 are listed as the last item in this Appendix. The temperature levels associated with voltage levels are exactly ten times the voltage output or one hundred time the machine unit output of the analog computer. Thus a voltage of 5.5 volts(.55 machine units) represents 55 degrees F. The setting of the individual room temperatures would be accomplished by putting the appropriate octal number into the appropriate slot in the program.

APPENDIX E
H-8 PROGRAM FLOW CHART



APPENDIX E



NOT-EXISTING PAGE
BY MISNUMBERING.

APPENDIX E
MNEMONIC CODE PROGRAM

```
040 100 MVI A 116
        OUT 277
        MVI A 005
        OUT 277
        MVI A 116
        OUT 271
        MVI A 005
        OUT 271

040 120 MVI A 000
        OUT 270
        IN 270
        CPI 060
        JC 300 040
        IN 270
        CPI 144
        JC 244 040
        IN 270
        CPI 255
        JC 210 040

040 151 MVI A (1)
        STA 100 041
        MVI A (2)
        STA 101 041
        MVI A (3)
        STA 102 041
        MVI A (4)
        STA 103 041
        MVI A (5)
        STA 104 041

        JMP 350 040

040 210 MVI A (1)
        STA 100 041
        MVI A (2)
        STA 101 041
        MVI A (3)
        STA 102 041
        MVI A (4)
        STA 103 041
        MVI A (5)
        STA 104 041

        JMP 350 040
```

APPENDIX E

```
040 244 MVI A (1)
        STA 100 041
        MVI A (2)
        STA 101 041
        MVI A (3)
        STA 102 041
        MVI A (4)
        STA 103 041
        MVI A (5)
        STA 104 041

        JMP 350 040

040 300 MVI A (1)
        STA 100 041
        MVI A (2)
        STA 101 041
        MVI A (3)
        STA 102 041
        MVI A (4)
        STA 103 041
        MVI A (5)
        STA 104 041

        JMP 350 040

040 350 LXI SP 110 041
        LXI B 064 041
        LXI D 071 041
        LXI H 100 041

040 364 LDAX B
        OUT 270
        IN 270
        CMP M
        JNC 004 041
        XTHL
        LDAX D
        ORA M
        STA 076 041
        XTHL

041 004 LDAX B
        CPI 005
        JZ 020 041
        INX B
        INX D
        INX H
        JMP 364 040

041 020 LDA 076 041
        OUT 276
        MVI A 000
        STA 076 041
        JMP 120 040
```

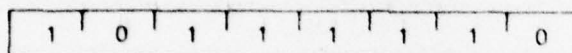
APPENDIX E

MNEMONIC CODE DEFINITIONS ⁷

276

CMP M (Compare memory) $(A) - ((H) (L))$

The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. **The Z flag is set to 1 if $(A) = ((H) (L))$. The CY flag is set to 1 if $(A) < ((H) (L))$.**



Cycles: 2

States: 7

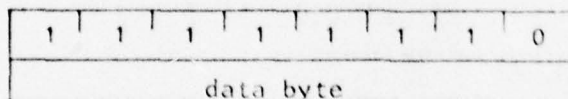
Addressing: reg. indirect

Flags: Z,S,P,CY,AC

376

CPI data (Compare immediate) $(A) - (\text{byte } 2)$

The content of the second byte of the instruction is subtracted from the content of the accumulator. The condition flags are set by the result of the subtraction. **The Z flag is set to 1 if $(A) = (\text{byte } 2)$. The CY flag is set to 1 if $(A) < (\text{byte } 2)$.**



Cycles: 2

States: 7

Addressing: immediate

Flags: Z,S,P,CY,AC

⁷ Heathkit Manual for the Digital Computer Model H-8 Operation 595-2014
(Benton Harbor, Michigan: Heath Company, 1977) pages 40-57.

APPENDIX E

CONDITION	CCC	OCTAL
NZ — not zero ($Z = 0$)	000	0
Z — zero ($Z = 1$)	001	1
NC — no carry ($CY = 0$)	010	2
C — carry ($CY = 1$)	011	3
PO — parity odd ($P = 0$)	100	4
PE — parity even ($P = 1$)	101	5
P — plus ($S = 0$)	110	6
M — minus ($S = 1$)	111	7

303

JMP addr (jump) $(PC) \leftarrow (\text{byte 3}) (\text{byte 2})$

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

1	1	0	0	0	0	1	1
low-order addr							
high-order addr							

Cycles: 3

States: 10

Addressing: immediate

Flags: none

3 (0-7)2

Jcondition addr (Condition jump)

If (CCC),

 $(PC) \leftarrow (\text{byte 3}) (\text{byte 2})$

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.

1	1	C	C	C	0	1	0
low-order addr							
high-order addr							

Cycles: 3

States: 10

Addressing: immediate

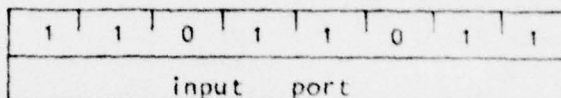
Flags: none

APPENDIX E

333

IN port (Input) $(A) \leftarrow \text{(data)}$

The data placed on the eight-bit bi-directional data bus by the specified port is moved to the accumulator.

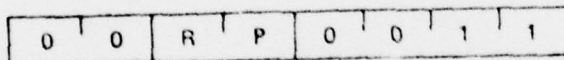


Cycles: 3
 States: 10
 Addressing: direct
 Flags: none

003	(B,C)	043	(H,L)
023	(D,E)	063	(S,P)

INX rp (Increment register pair) $(rh) (rl) \leftarrow (rh) (rl) + 1$

The content of the register pair rp is incremented by one. NOTE: No condition flags are affected.

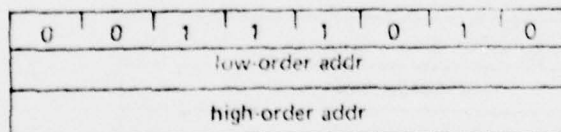


Cycles: 1
 States: 5
 Addressing: register
 Flags: none

072

LDA addr (Load Accumulator direct) $(A) \leftarrow \text{(byte 3) (byte 2)}$

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to the accumulator.



Cycles: 4
 States: 13
 Addressing: direct
 Flags: none

APPENDIX E

012 (B, C) 032 (D, E)

LDAX rp (Load accumulator indirect) $\{A\} \leftarrow \{[rp]\}$

The content of the memory location, whose address is in the register pair *rp*, is moved to register A. NOTE: Only register pairs *rp* = B (registers B and C) or *rp* = D (registers D and E) may be specified.

0	0	R	P	1	0	1	0
---	---	---	---	---	---	---	---

Cycles: 2

States: 7

Addressing: reg. indirect

Flags: none

001 (B, C)

041 (H, L)

021 (D, E)

061 (S, P)

LXI rp, data 16 (Load register pair immediate) $\{rh\} \leftarrow \{\text{byte 3}\}$ $\{rl\} \leftarrow \{\text{byte 2}\}$

Byte 3 of the instruction is moved into the high-order register (*rh*) of the register pair *rp*. Byte 2 of the instruction is moved into the low-order register (*rl*) of the register pair *rp*.

0	0	R	P	0	0	0	1
low-order data							
high-order data							

Cycles: 3

States: 10

Addressing: immediate

Flags: none

0 (0-7)6

MVI r, data (Move immediate) 0(0-7)6 $\{r\} \leftarrow \{\text{byte 2}\}$

The content of byte 2 of the instruction is moved to register *r*.

0	0	D	D	D	1	1	0
data byte							

Cycles: 2

States: 7

Addressing: immediate

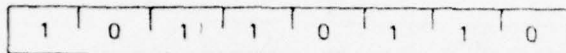
Flags: none

APPENDIX E

266

ORA M (OR memory) $(A) \leftarrow (A) \vee ((H) (L))$

The content of the memory location whose address is contained in the H and L registers is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.



Cycles: 2

States: 7

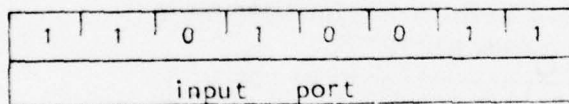
Addressing: reg. indirect

Flags: Z,S,P,CY,AC

323

OUT port (Output) $(data) \leftarrow (A)$

The content of the accumulator is placed on the eight-bit bi-directional data bus for transmission to the specified port.



Cycles: 3

States: 10

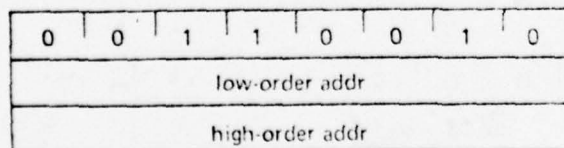
Addressing: direct

Flags: none

062

STA addr (Store Accumulator direct) $((\text{byte } 3) (\text{byte } 2)) \leftarrow (A)$

The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.



Cycles: 4

States: 13

Addressing: direct

Flags: none

APPENDIX E

343

XTHL (Exchange stack top with H and L) $(L) \leftrightarrow ((SP))$ $(H) \leftrightarrow ((SP) + 1)$

The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

1	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

Cycles: 5

States: 18

Addressing: reg. indirect

Flags: none

APPENDIX E
OCTAL CODE PROGRAM LISTING

040	100	076	116	323	277
	104	076	005	323	277
	110	076	116	323	271
	114	076	005	323	271
	120	076	000	323	270
	124	333	270	376	060
	130	332	300	040	333
	134	270	376	144	332
	140	244	040	333	270
	144	376	255	332	210
	150	040	076	(1)	062
	154	100	041	076	(2)
	160	062	101	041	076
	164	(3)	062	102	041
	170	076	(4)	062	103
	174	041	076	(5)	062
040	200	104	041	303	350
	204	040	000	000	000
	210	076	(1)	062	100
	214	041	076	(2)	062
	220	101	041	076	(3)
	224	062	102	041	076
	230	(4)	062	103	041
	234	076	(5)	062	104
	240	041	303	350	040
	244	076	(1)	062	100
	250	041	076	(2)	062
	254	101	041	076	(3)
	260	062	102	041	076
	264	(4)	062	103	041
	270	076	(5)	062	104
	274	041	303	350	040
040	300	076	(1)	062	100
	304	041	076	(2)	062
	310	101	041	076	(3)
	314	062	102	041	076
	320	(4)	062	103	041
	324	076	(5)	062	104
	330	041	303	350	040
	334	XXX	XXX	XXX	XXX
	340	XXX	XXX	XXX	XXX
	344	XXX	XXX	XXX	XXX

APPENDIX E

	350	061	110	041	001
	354	064	041	021	071
	360	041	041	100	041
	364	012	323	270	333
	370	270	276	322	004
	374	041	343	032	266
041	000	062	076	041	343
	004	012	376	005	312
	010	020	041	003	023
	014	043	303	364	040
	020	072	076	041	323
	024	276	076	000	062
	030	076	041	303	120
	034	040	XXX	XXX	XXX
	040	XXX	XXX	XXX	XXX
	044	XXX	XXX	XXX	XXX
	050	XXX	XXX	XXX	XXX
	054	XXX	XXX	XXX	XXX
	060	XXX	XXX	XXX	XXX
	064	001	002	003	004
	070	005	001	002	004
	074	010	020	XXX	XXX
041	100	(1)	(2)	(3)	(4)
	104	(5)	XXX	XXX	XXX

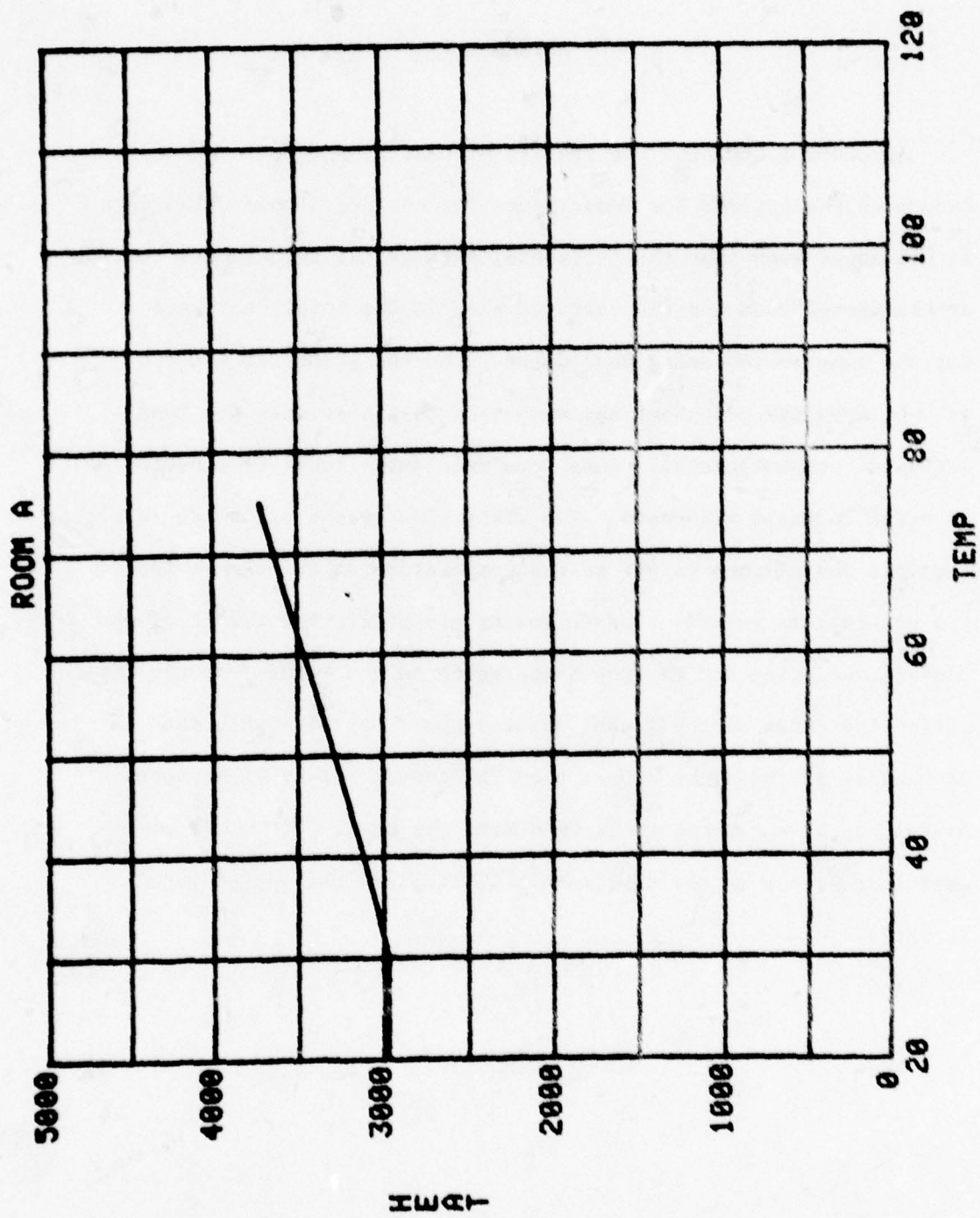
APPENDIX E
ANALOG VOLTAGE TO OCTAL CONVERSION TABLE

9.96	377	8.16	322	6.36	245	4.56	170	2.76	113
9.92	376	8.12	321	6.32	244	4.52	167	2.72	112
9.88	375	8.08	320	6.28	243	4.48	166	2.68	111
9.84	374	8.04	317	6.24	242	4.44	165	2.64	110
9.80	373	8.00	316	6.20	241	4.40	164	2.60	107
9.76	372	7.96	315	6.16	240	4.36	163	2.56	106
9.72	371	7.92	314	6.12	237	4.32	162	2.52	105
9.68	370	7.88	313	6.08	236	4.28	161	2.48	104
9.64	367	7.84	312	6.04	235	4.24	160	2.44	103
9.60	366	7.80	311	6.00	234	4.20	157	2.40	102
9.56	365	7.76	310	5.96	233	4.16	156	2.36	101
9.52	364	7.72	307	5.92	232	4.12	155	2.32	100
9.48	363	7.68	306	5.88	231	4.08	154	2.28	077
9.44	362	7.64	305	5.84	230	4.04	153	2.24	076
9.40	361	7.60	304	5.80	227	4.00	152	2.20	075
9.36	360	7.56	303	5.76	226	3.96	151	2.16	074
9.32	357	7.52	302	5.72	225	3.92	150	2.12	073
9.28	356	7.48	301	5.68	224	3.88	147	2.08	072
9.24	355	7.44	300	5.64	223	3.84	146	2.04	071
9.20	354	7.40	277	5.60	222	3.80	145	2.00	070
9.16	353	7.36	276	5.56	221	3.76	144	1.96	067
9.12	352	7.32	275	5.52	220	3.72	143	1.92	066
9.08	351	7.28	274	5.48	217	3.68	142	1.88	065
9.04	350	7.24	273	5.44	216	3.64	141	1.84	064
9.00	347	7.20	272	5.40	215	3.60	140	1.80	063
8.96	346	7.16	271	5.36	214	3.56	137	1.76	062
8.92	345	7.12	270	5.32	213	3.52	136	1.72	061
8.88	344	7.08	267	5.28	212	3.48	135	1.68	060
8.84	343	7.04	266	5.24	211	3.44	134	1.64	057
8.80	342	7.00	265	5.20	210	3.40	133	1.60	056
8.76	341	6.96	264	5.16	207	3.36	132	1.56	055
8.72	340	6.92	263	5.12	206	3.32	131	1.52	054
8.68	337	6.88	262	5.08	205	3.28	130	1.48	053
8.64	336	6.84	261	5.04	204	3.24	127	1.44	052
8.60	335	6.80	260	5.00	203	3.20	126	1.40	051
8.56	334	6.76	257	4.96	202	3.16	125	1.36	050
8.52	333	6.72	256	4.92	201	3.12	124	1.32	047
8.48	332	6.68	255	4.88	200	3.08	123	1.28	046
8.44	331	6.64	254	4.84	177	3.04	122	1.24	045
8.40	330	6.60	253	4.80	176	3.00	121	1.20	044
8.36	327	6.56	252	4.76	175	2.96	120	1.16	043
8.32	326	6.52	251	4.72	174	2.92	117	1.12	042
8.28	325	6.48	250	4.68	173	2.88	116	1.08	041
8.24	324	6.44	247	4.64	172	2.84	115	1.04	040
8.20	323	6.40	246	4.60	171	2.80	114	1.00	037

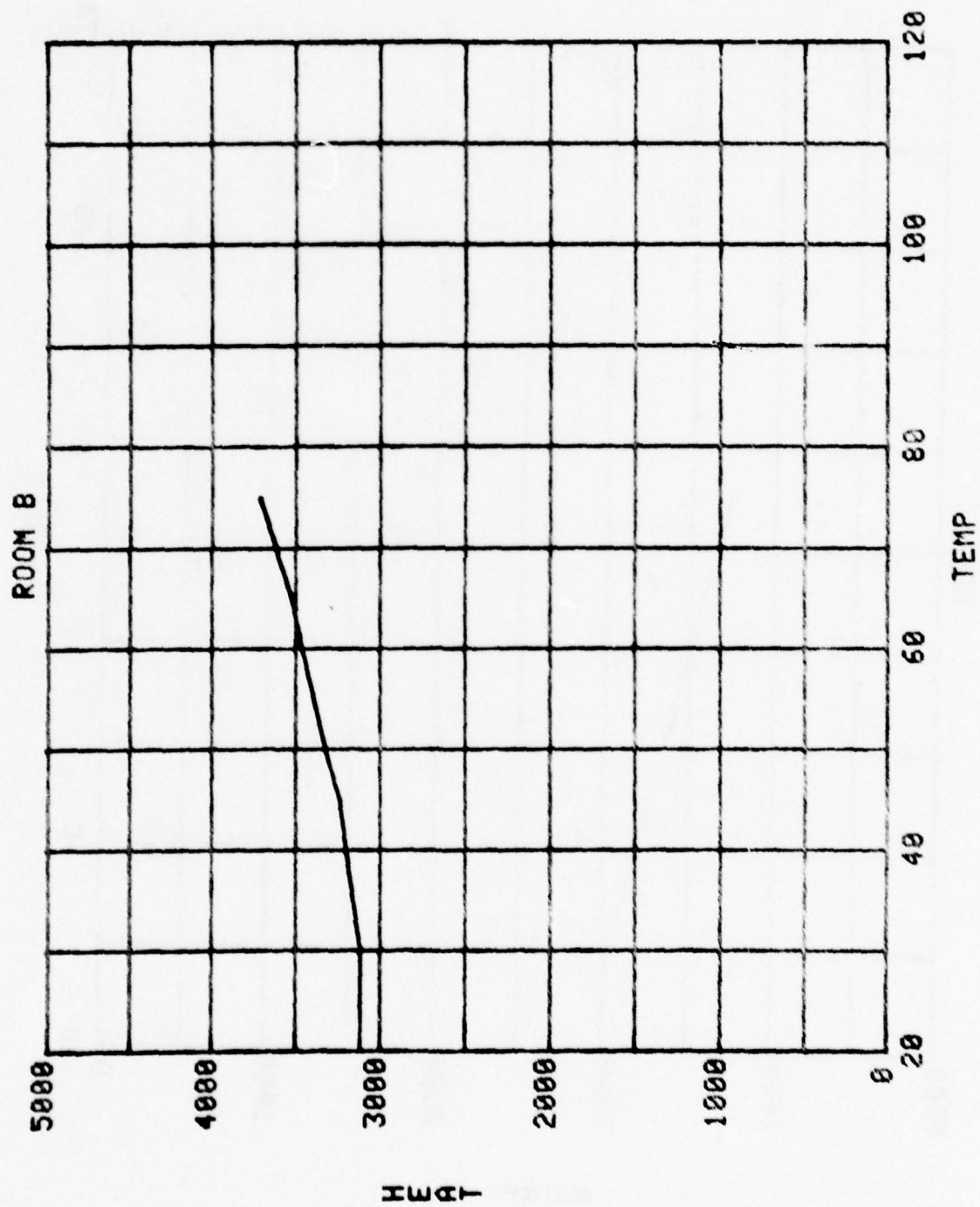
APPENDIX F

Appendix F contains the results of test runs made in order to determine the optimum low temperature for an unused room. The data is presented such that the horizontal axis is the temperature setting of the unused room and the vertical axis is the total heat used for the time period being considered. The bar graphs at the end of the Appendix represent the amount of heat used when the rooms indicated are not heated. Runs were made for a total time period of eight hours in all cases. The first nine graphs are those resulting when the unused room is set to the temperature on the x-axis for the entire time period. The following graphs are the result of the unused room being set to the temperature on the x-axis for the time period indicated on the graph, first 3 hours out of eight, then 45 minutes out of eight hours, then 23 minutes out of eight hours. As the graphs show, the trend is always the same. The total heat used increases when the temperature setting of the unused room is raised.

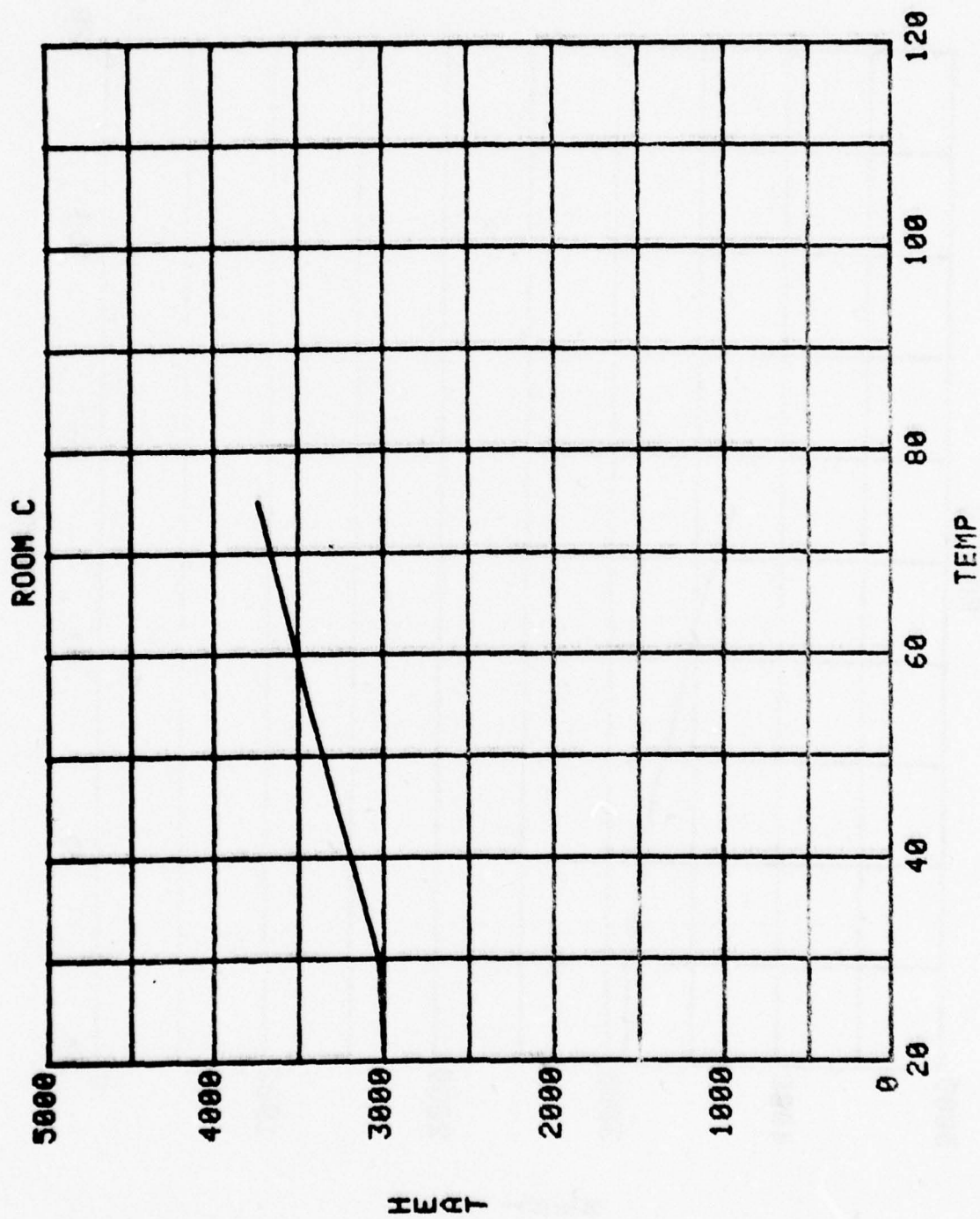
APPENDIX F



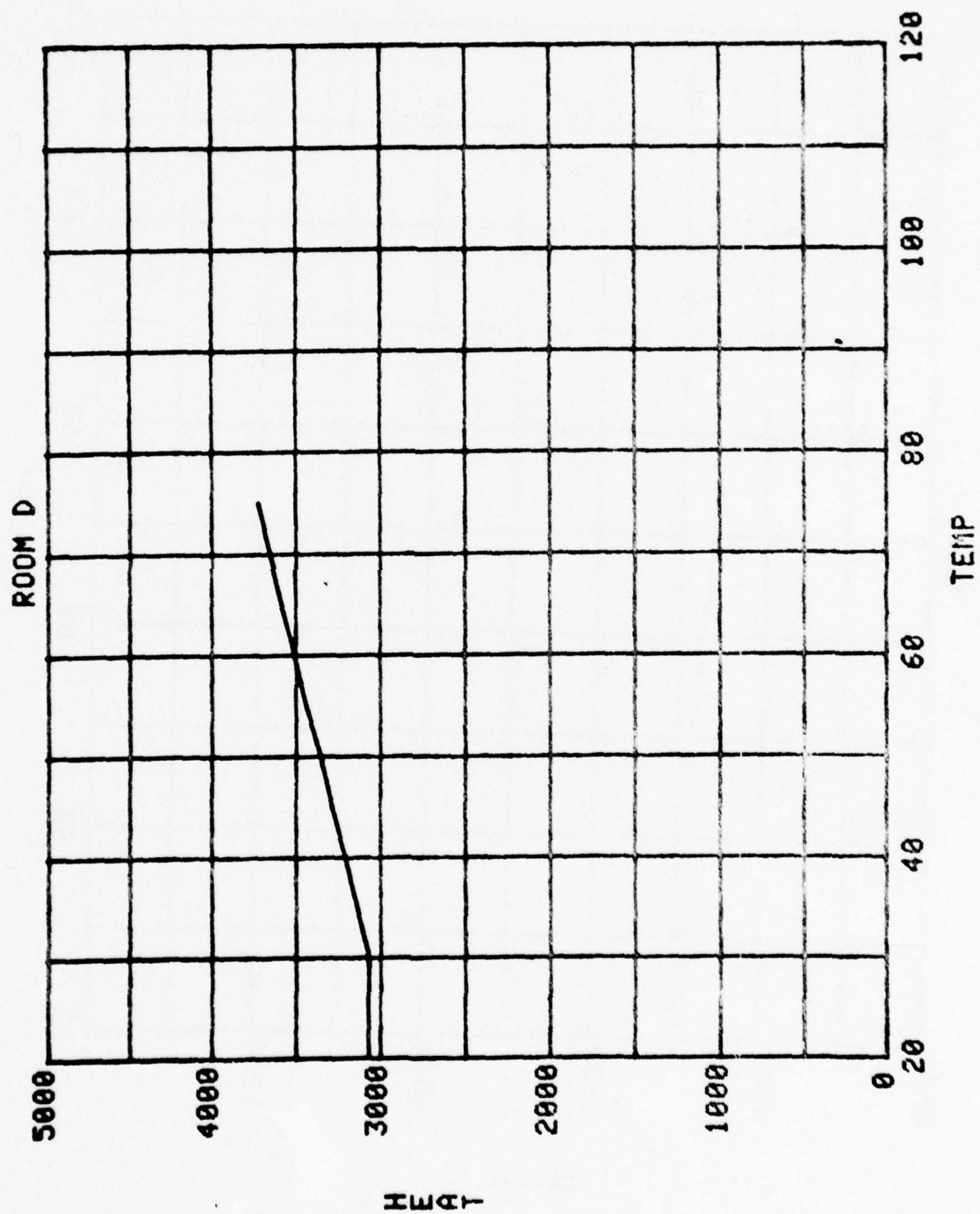
APPENDIX F



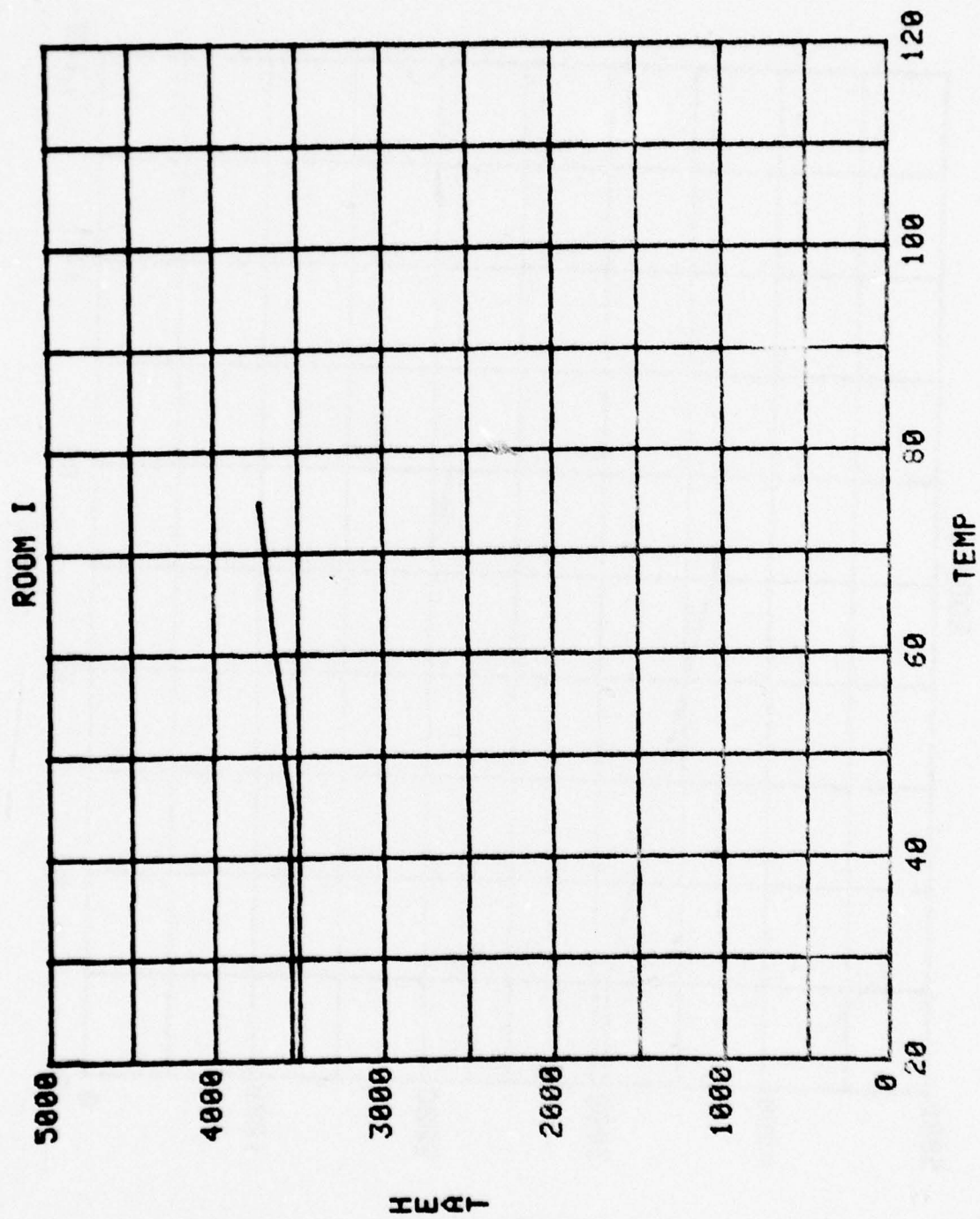
APPENDIX F



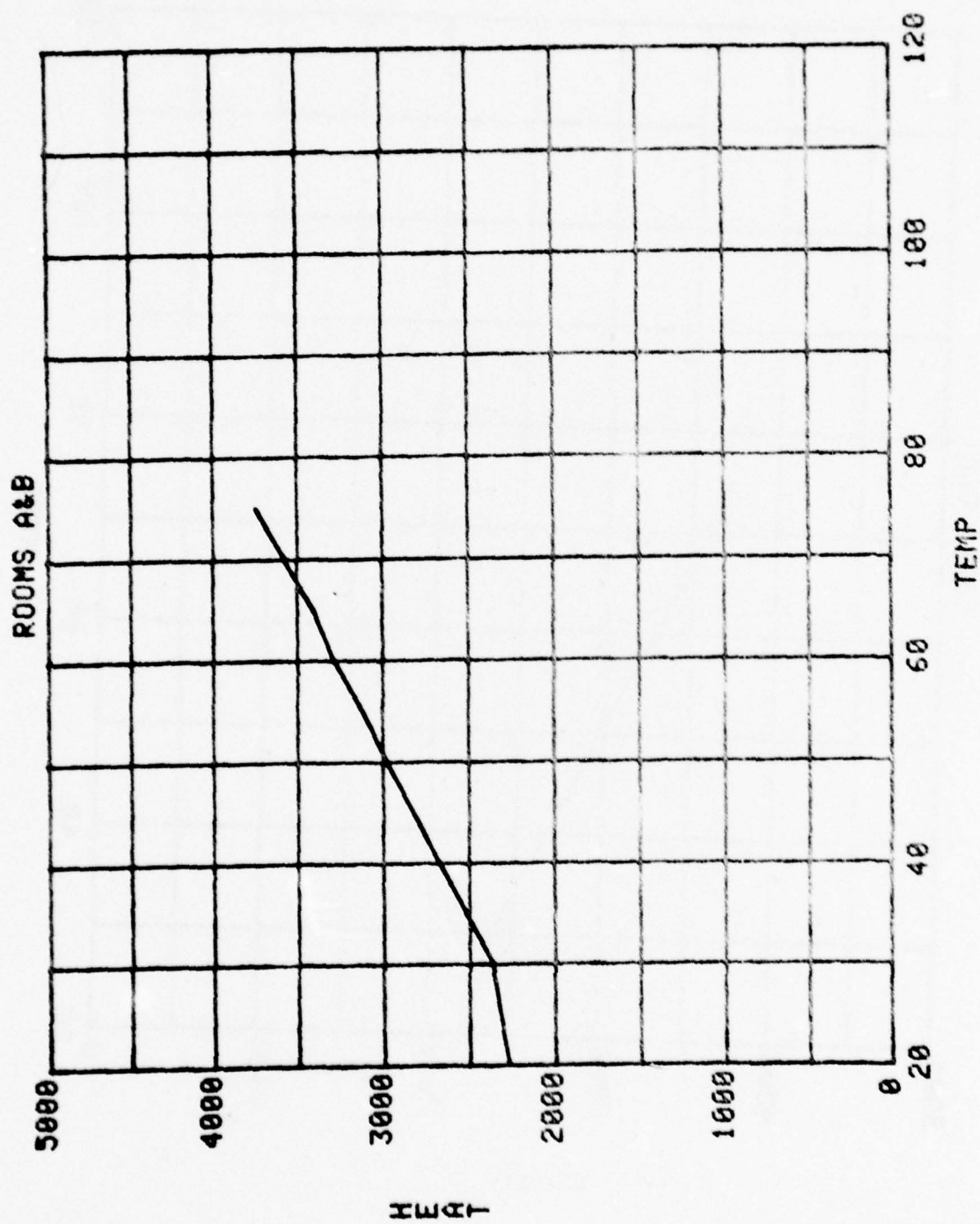
APPENDIX F



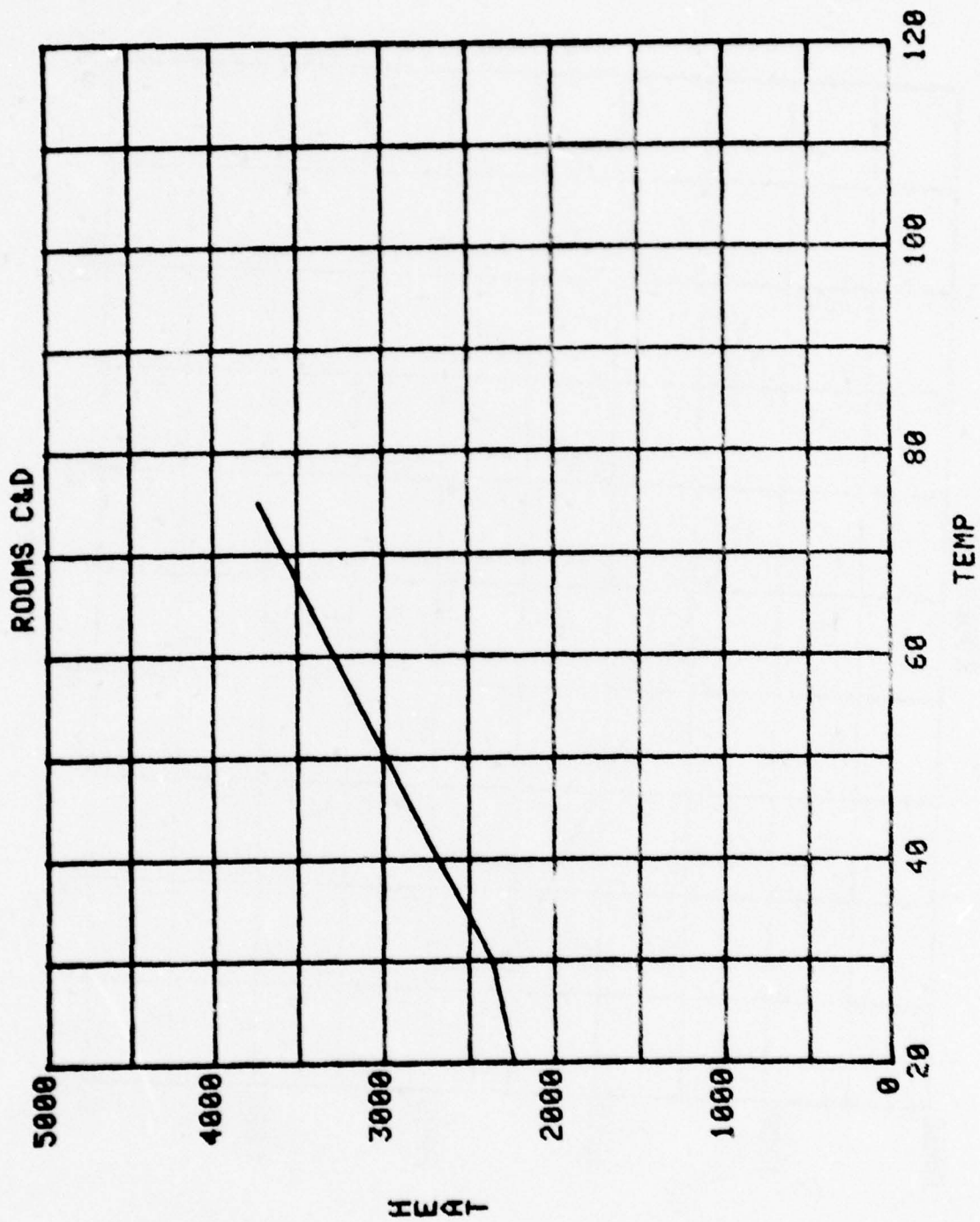
APPENDIX F



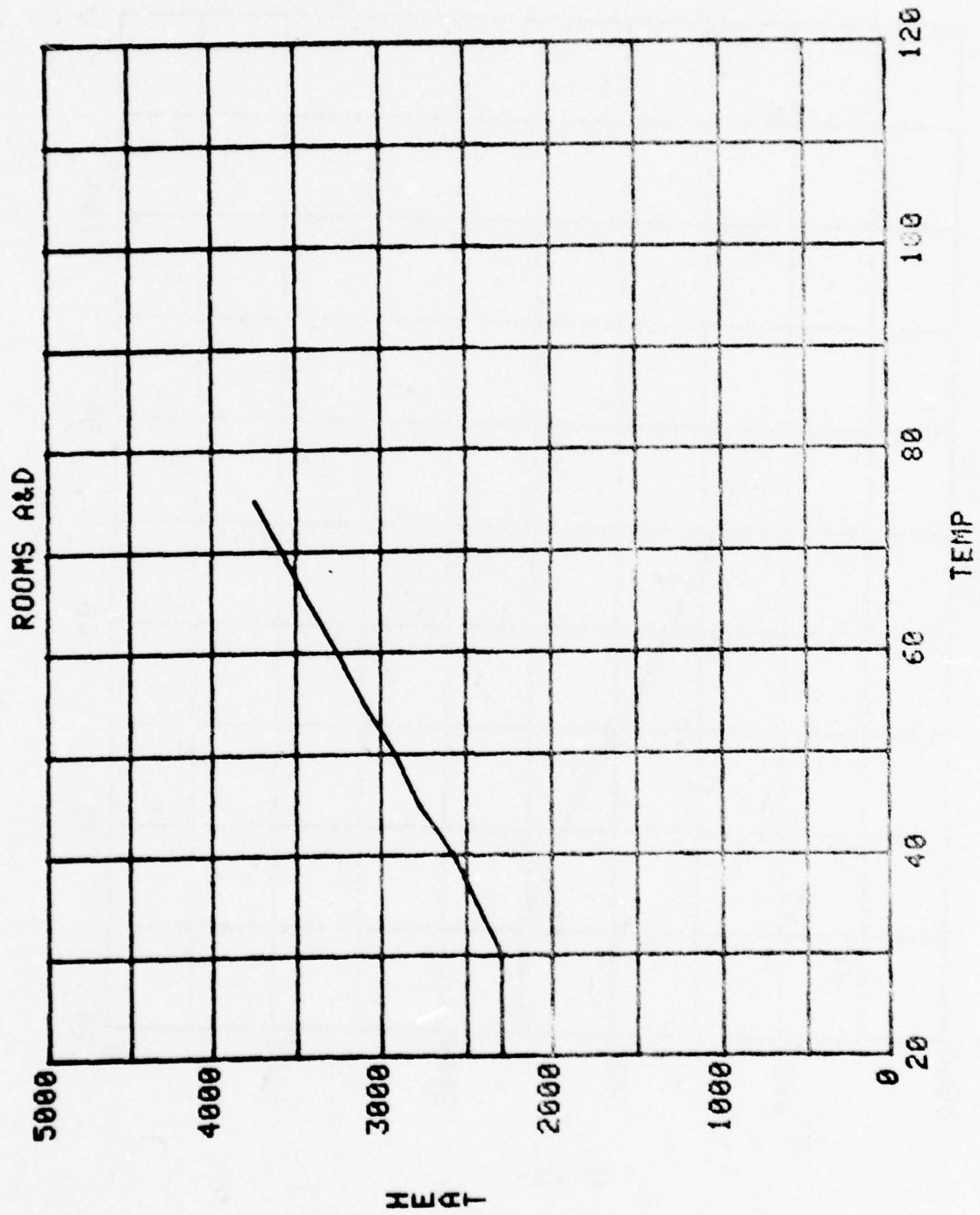
APPENDIX F



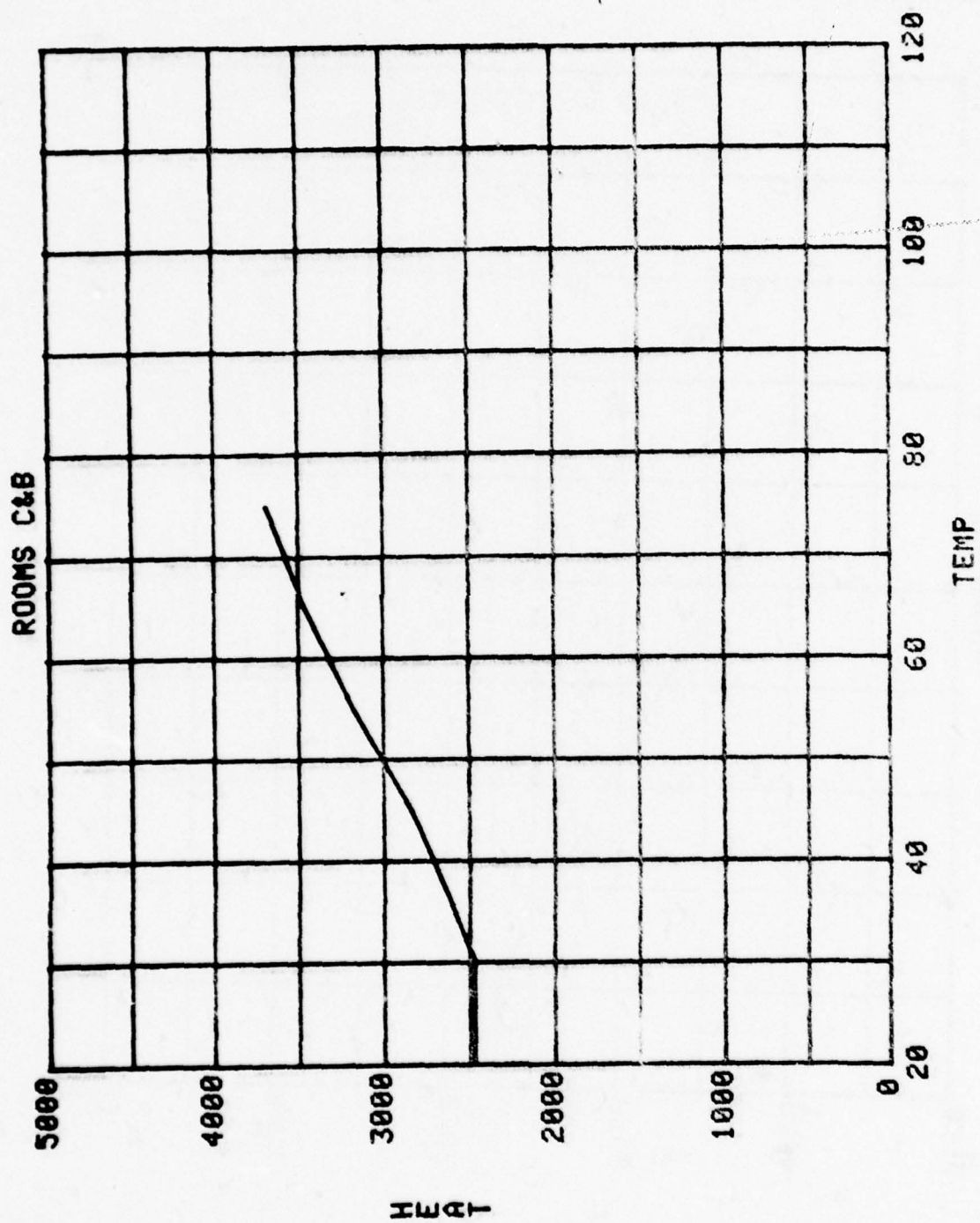
APPENDIX F



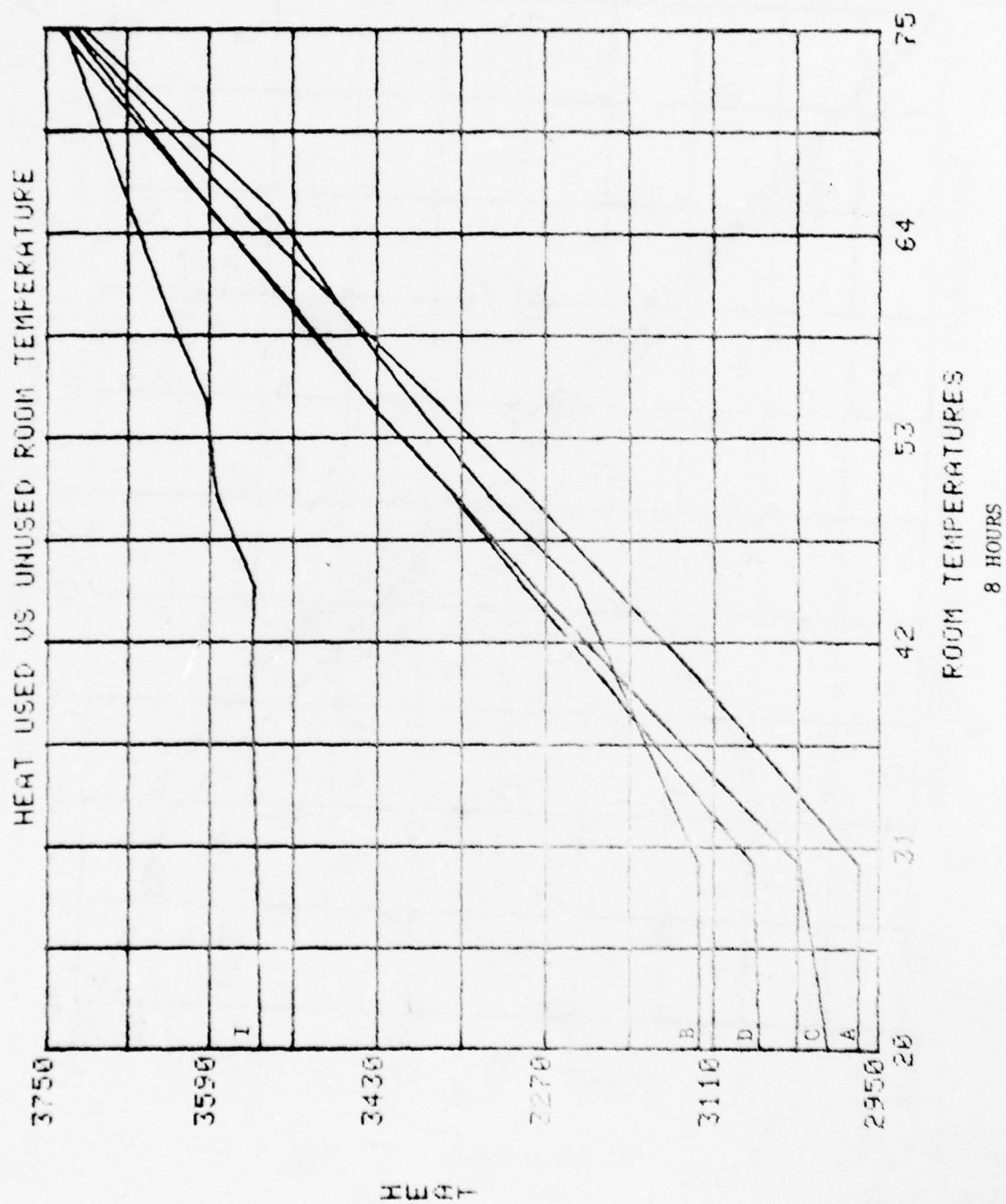
APPENDIX F



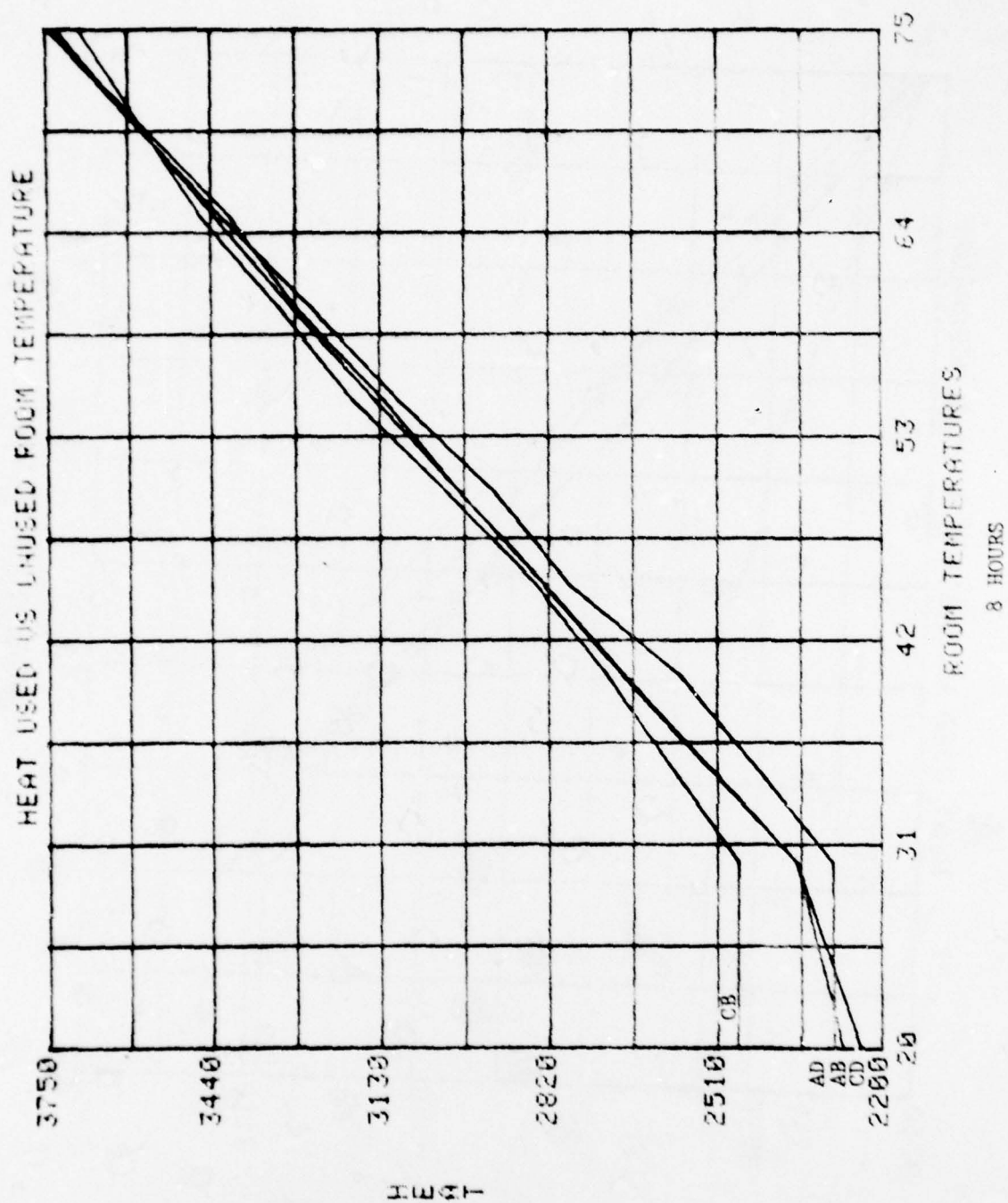
APPENDIX F



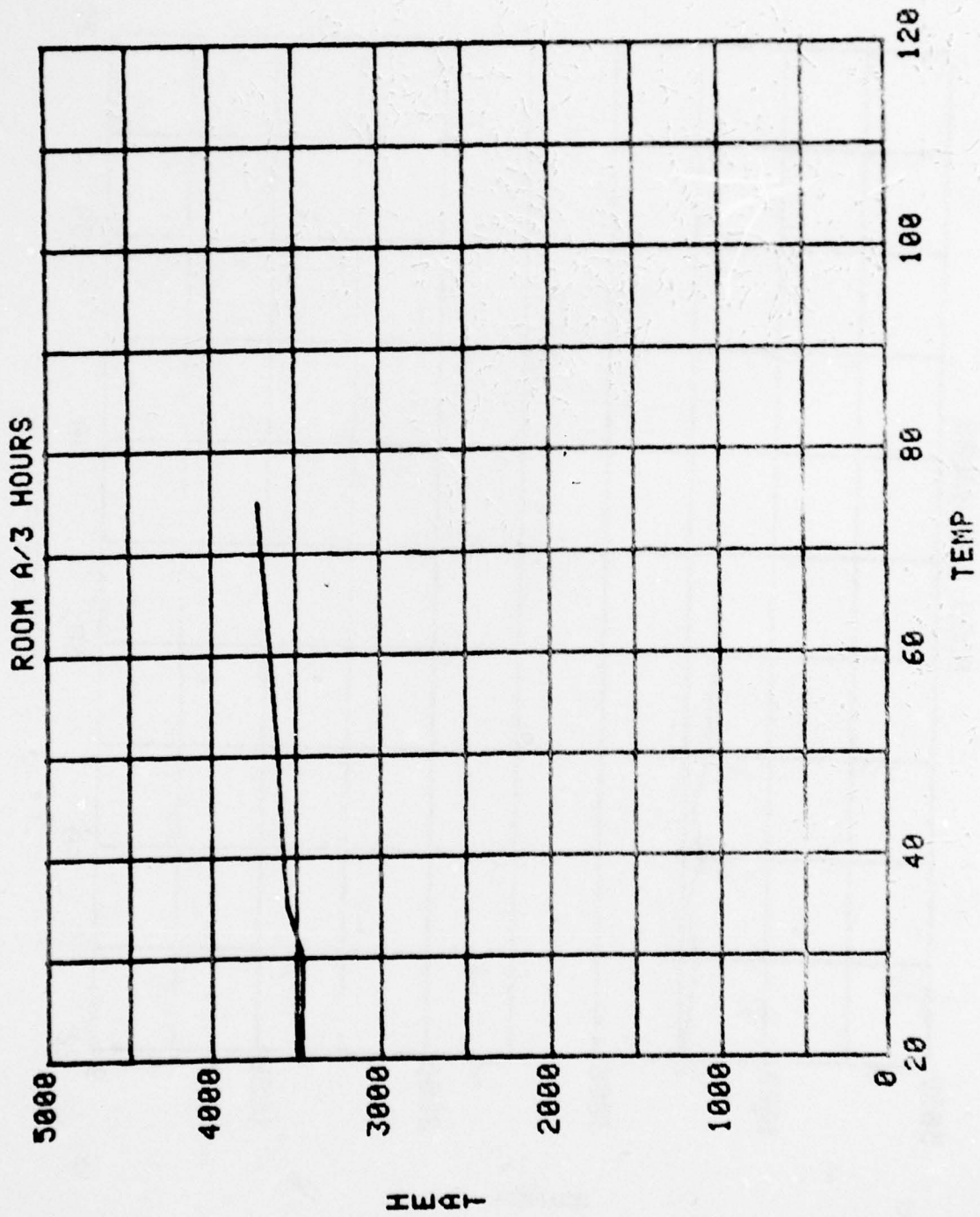
APPENDIX F



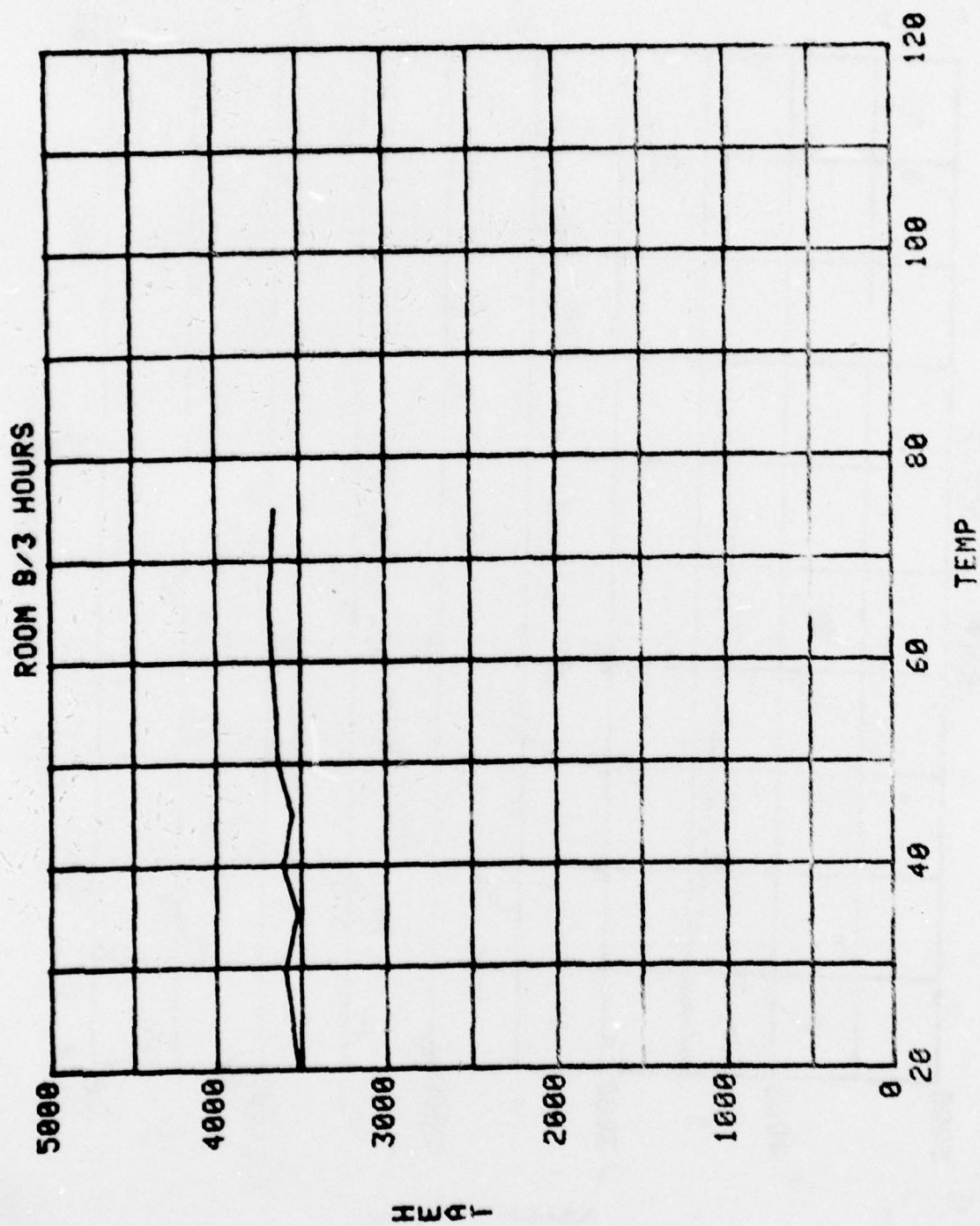
APPENDIX F



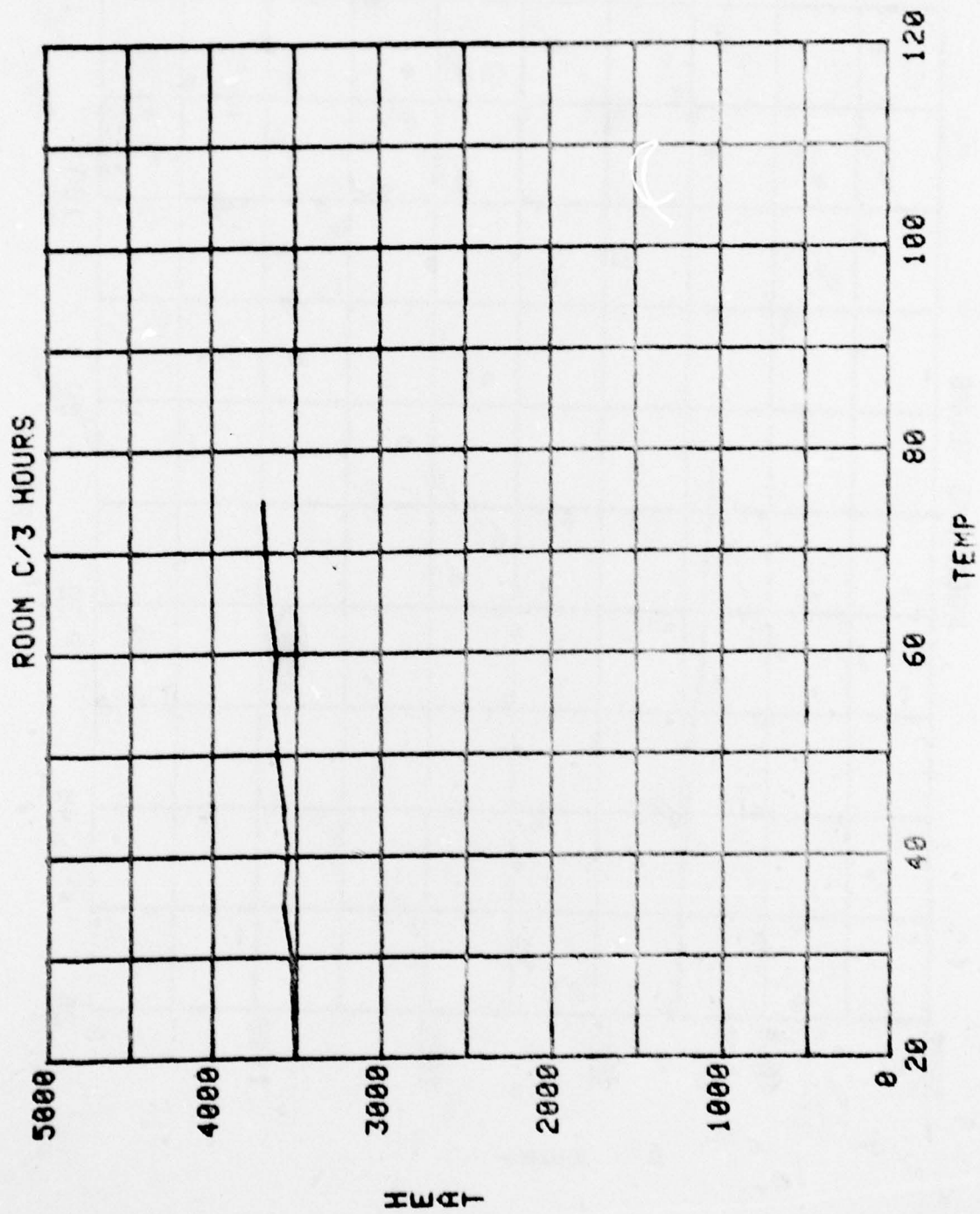
APPENDIX F



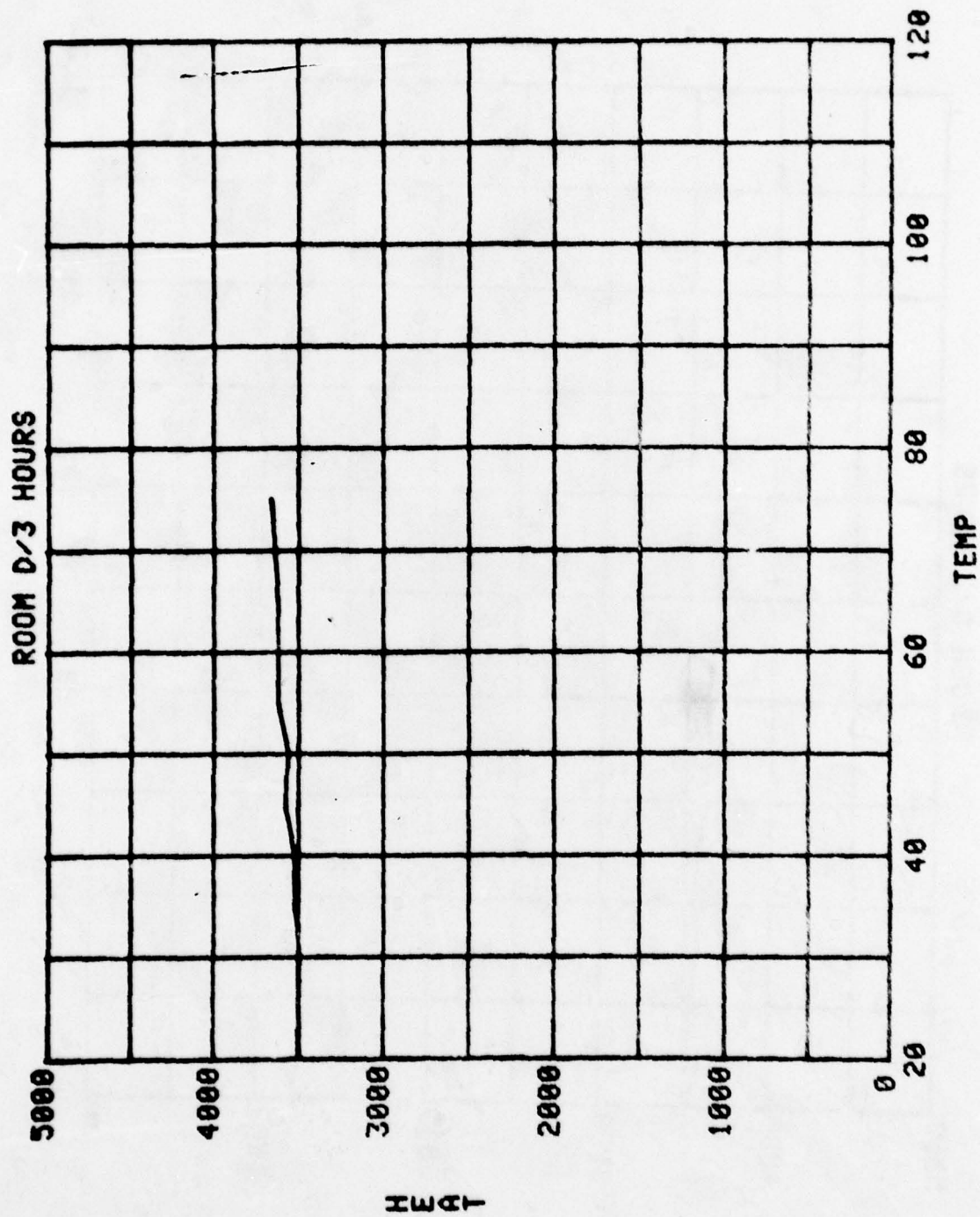
APPENDIX F



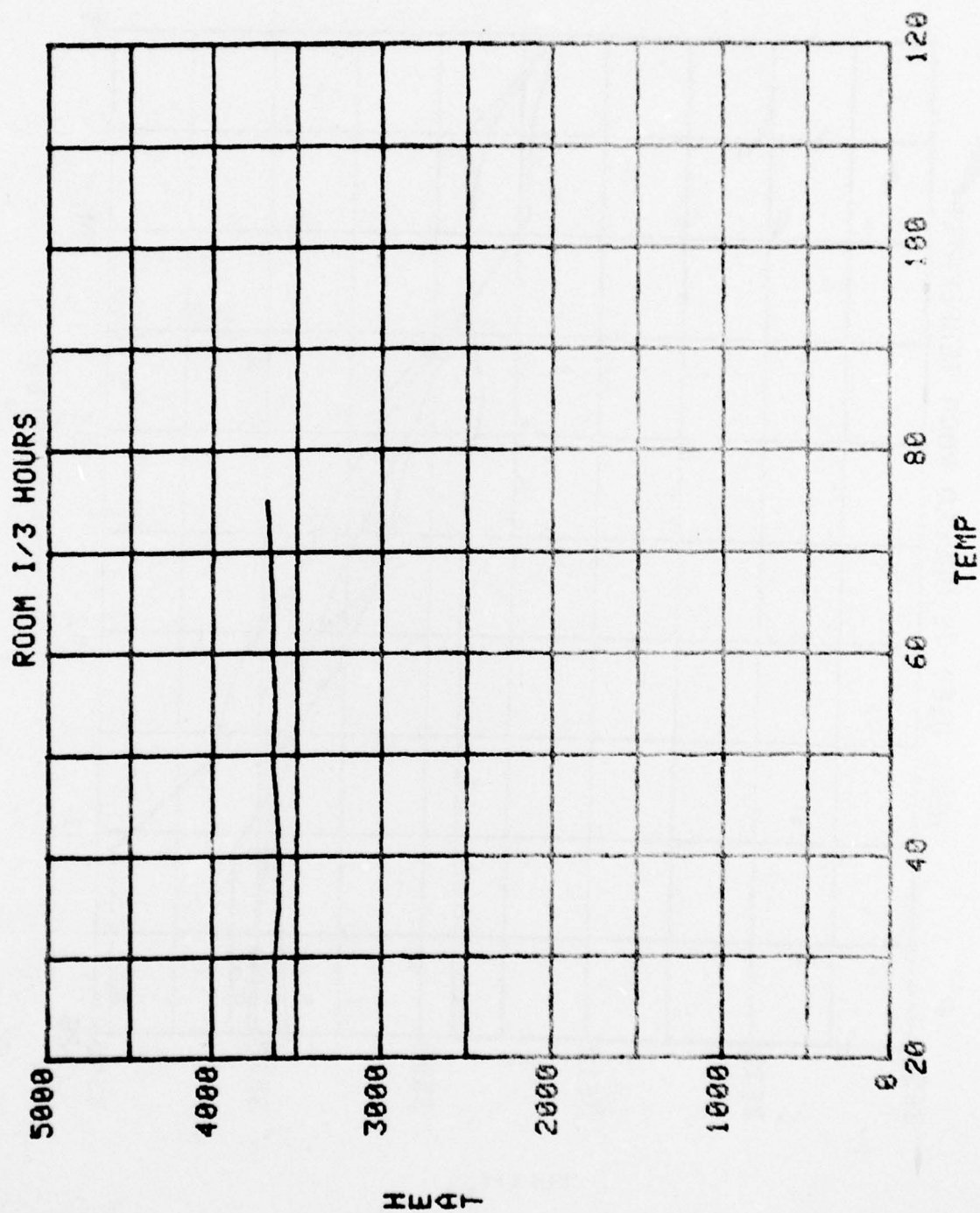
APPENDIX F



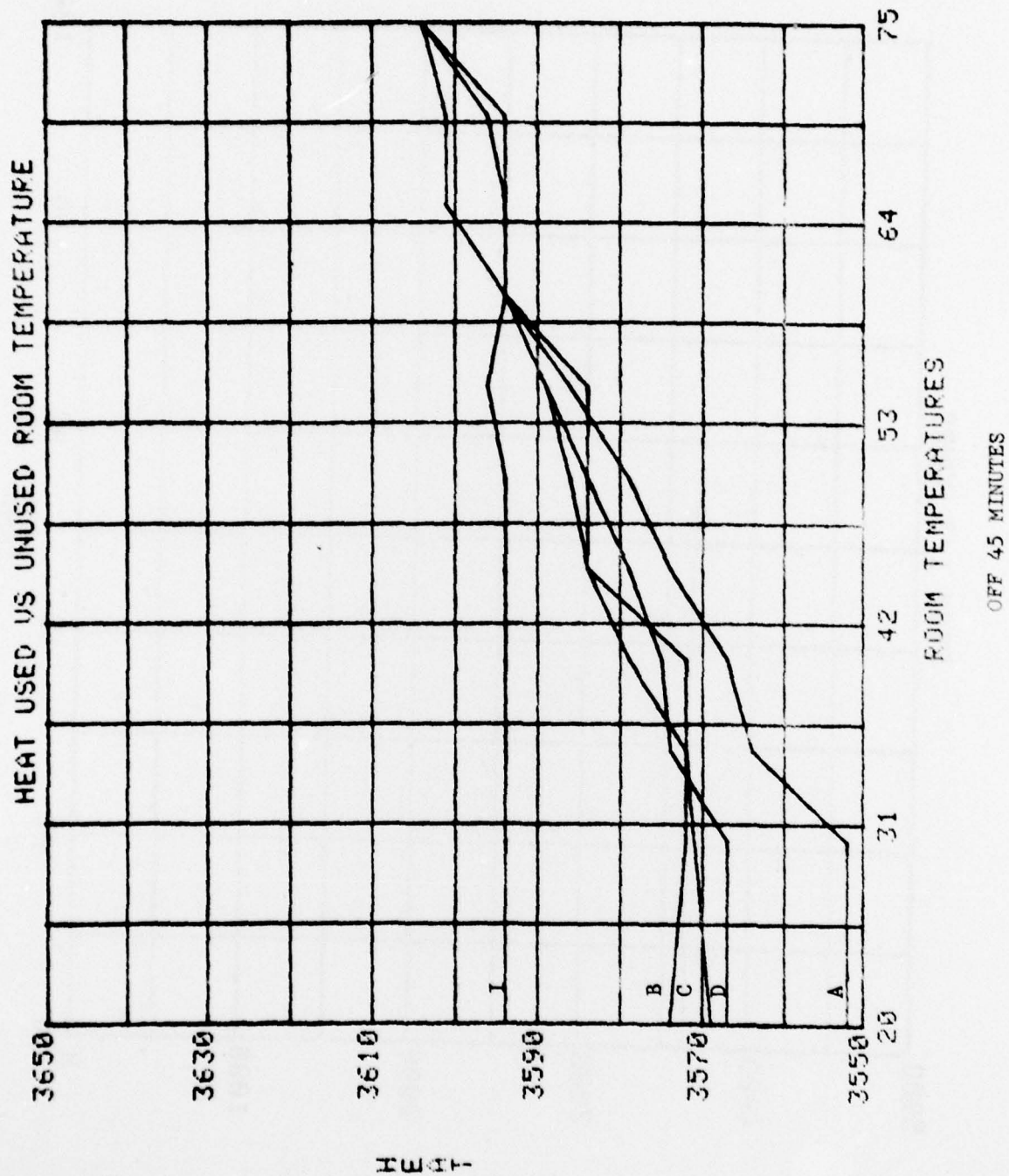
APPENDIX F



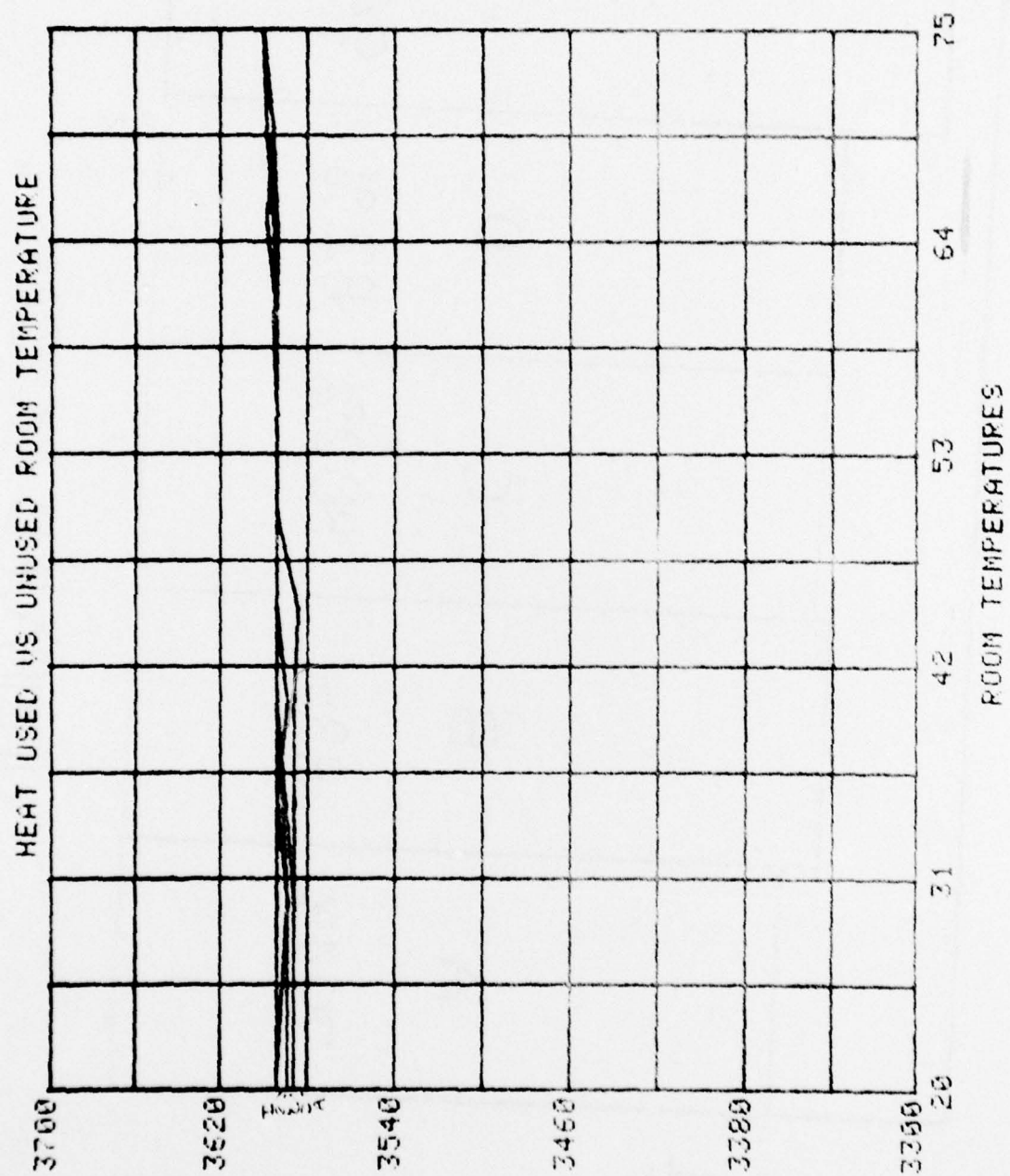
APPENDIX F



APPENDIX F



APPENDIX F



OFF 23 MINUTES

APPENDIX F

% TOTAL HEAT				
	A	B	C	D
	79.4%	83.5%	80.2%	82.0%
				I
				94.7%

UNHEATED ROOM

APPENDIX F

% TOTAL HEAT	A+B	60.8%
	C+D	59.9%
	A+D	61.3%
	C+B	66.1%

UNHEATED ROOMS

APPENDIX G

This Appendix contains the use charts used to write sample programs for the digital controller. The first use chart represents the use habits of a family which does not use their house in the day, and does not use the bedrooms until retiring. The second use chart represents the use habits of a family which does use the house during the day. The charts represent typical habits for people and do not really mean that there is actually a family which follows these use charts. The results obtained from using the charts is meant to be representative of what could be achieved with the controller.

APPENDIX G

USE CHART I

TIME ROOM	0700	0830	1700	2300	0700
	A	B	C	D	I
HEATED	UNHEATED	HEATED	UNHEATED	UNHEATED	UNHEATED
HEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED
HEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED
HEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED
HEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED	UNHEATED

APPENDIX G

USE CHART II

TIME ROOM	0800				
	0800	1200	1700	2300	0800
A	HEATED	HEATED	HEATED	HEATED	UNHEATED
B	HEATED	HEATED	HEATED	HEATED	UNHEATED
C	UNHEATED	UNHEATED	UNHEATED	HEATED	HEATED
D	UNHEATED	UNHEATED	UNHEATED	UNHEATED	HEATED
I	HEATED	HEATED	HEATED	HEATED	HEATED

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER U.S.N.A. - TSPR; no. 92 (1978)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MICROPROCESSOR-BASED DIGITAL CONTROL OF ENVIRONMENTAL SYSTEMS.	5. TYPE OF REPORT & PERIOD COVERED Final, 1977/78.	
7. AUTHOR(s) Eddy D. Kee.	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS United States Naval Academy, Annapolis, Md.	8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS United States Naval Academy, Annapolis, Md.	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE 1 June 1978.	
	13. NUMBER OF PAGES 146.	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED.	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) This document has been approved for public release; its distribution is UNLIMITED.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) This document has been approved for public release; its distribution is UNLIMITED.		
18. SUPPLEMENTARY NOTES Accepted by the chairman of the Trident Scholar Committee.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Heating. Automatic control. Environmental engineering (Buildings) Energy conservation. Buildings.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This project investigated the use of microprocessors and associated hardware to control the heating of a building in order to save energy. A house was simulated on the hybrid computer and controlled by a microprocessor-based digital controller. The work includes the mathematical model, simulation, flow charts, computer programs for controller, etc. to use for different type of houses and insulations. Thests showed that a significant amount of energy was saved by using the controller, as much as 30-50 o/o, depending on the use-habits of the house.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)